Optimal buffer control during congestion in an ATM network node *

Leandros Tassiulas Yaochung Hung
Shivendra S. Panwar
Department of Electrical Engineering and
Center for Advanced Technology in Telecommunications
Polytechnic University, Brooklyn, NY 11201 †

Abstract

In this paper we study the problem of optimal buffer space priority control in an ATM network node. The buffer of a transmission link is shared among the cells of several traffic classes waiting for transmission through the link. When the number of cells to be stored in the buffer exceeds the available buffer space, certain cells have to be dropped. Different traffic classes have different sensitivities to cell losses. By appropriately selecting the classes of cells which are dropped in case of overflow, we can have the more sensitive classes suffer smaller cell losses. Arriving cells might be blocked from entering the system or they may be dropped after they are already in the buffer. Depending on the control that we have on the system, three classes of policies are distinguished. In each one, policies that schedule the buffer allocation in some optimal manner are identified.

1. Introduction

One of the main problems arising in the area of high speed communication networks is the design of control algorithms for the efficient sharing of the buffer space in an ATM node. Cells of different traffic types arrive at the node and are stored in a buffer until their transmission. Cells of different types may be generated by a leaky bucket policing function which marks excessive traffic cells at the source network interface or by an encoding scheme which creates cells with different priorities[6]. When a cell finds the buffer full upon arrival, it may be discarded before admission into the system. The cell loss due to buffer overflow incurs a degradation in the overall system performance which is highly dependent on the type of the discarded cells. Certain traffic types are more sensitive to potential cell losses than others. We can reduce the probability of discarding a loss-sensitive cell due to buffer overflow if we block the admission of less loss sensitive cells. We may also consider expelling less loss sensitive cells from the buffer. In this paper we study how we can do this in an optimal manner.

We consider a single outgoing link and the corresponding dedicated buffer in a network node. The system is modeled by a single server queue(Figure 1). The queue has a buffer that can store B cells; this is called the *main buffer* in the following. Time is slotted and the transmission of a cell takes one slot. During one slot at most B_T cells may arrive to the system and they are placed in the *temporary buffer* which has length B_T. These cells may belong to different traffic types. This assumption is consistent with the structure of knockout-type ATM switches[9] or a switch with output queueing. At the end of each slot the cells from the temporary buffer are either placed in the main buffer or dropped from the system. Depending on the available control we have over the dropping of cells from the temporary or the main buffer and over the placement of the cells in the main buffer, we will distinguish three classes of policies. In all the policies considered it is assumed that the cells which enter the main buffer in every slot should join the end of the queue and rearrangement of cells is not allowed. Hence the FIFO discipline is preserved and the cells are delivered in order. This property is essential in virtual circuit connections.

The first class is that of *discarding policies.* A discarding policy cannot modify the state of the cells which are already in the main buffer. It can only control the admission of the cells from the temporary buffer, by blocking some if necessary, and the place-

*This work appeared in part in the Proceedings of the 1992 Conference on Information Sciences and Systems, Princeton, NJ, March 18-20, 1992.

†This work was supported by the Center for Advanced Technology in Telecommunications, Polytechnic University and by the NSF under grants NCR-8909719 and NCR-9115864.

0743-166X/93 $03.00 © 1993 IEEE

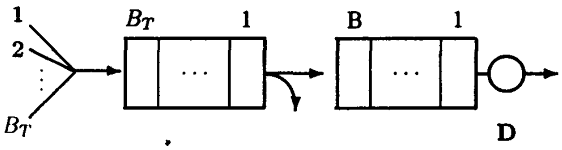


Figure 1: The system model

ment of the admitted cells in the main buffer. We show that the optimal discarding policy is of "multi-threshold type." That is, for each priority class there is a threshold, and if the number of cells in the main buffer exceeds that threshold, the cells of that class are blocked from admission. The policy is optimal in the sense that it minimizes the long run average blocking cost where a cost is associated with each cell that reflects the loss sensitivity of its class.

The second class of policies considered are the *pushout policies.* A pushout policy is allowed to expel cells from the main buffer in order to make space for cells in the temporary buffer which cannot enter the main buffer because it is full. A cell from the temporary buffer cannot be blocked from admission to the main buffer if there is space in the main buffer. We obtain the optimal pushout policy, which we call the *squeeze-out policy*, in a system with two priority classes. That policy places the cells in the main buffer, high priority first. If the buffer is full and there are cells in the temporary buffer, then the second priority cells are pushed out of the main buffer starting from those closest to the head of the queue. Notice that second priority cells are dropped to make space for other second priority cells that are appended to the end of the queue. The squeeze-out policy minimizes the blocking probability of the high priority(loss sensitive) class among all pushout policies.

The third class of policies considered are the *expelling policies.* Expelling policies are allowed to discard cells from the main buffer or block cells in the temporary buffer from admission into the main buffer irrespective of the system state. Properties of the optimal expelling policy are obtained that narrow down the set of candidate policies considerably in a system with two classes. More specifically we show that the cells are placed in the main buffer high priority first and low priority cells are pushed out, starting from the head of the queue, if there is no space like in the case of the squeeze-out policy. In addition to that, the optimal expelling policy may drop low priority cells even if the main buffer is not full but only if the low priority cell(s) dropped is(are) at the head of the queue.

Clearly, an expelling policy has more control over the system than discarding and pushout policies. In other words the class of expelling policies contains the discarding and pushout policies as subclasses. Policies of different classes have different degrees of implementation difficulty. For one approach that allows for the implementation of some of the policies considered in this paper, see [7].

The problem of sharing the buffer space among several competing traffic streams has attracted considerable attention in the past. Several strategies for buffer sharing, called space priority access methods, have been proposed and analyzed. Petr and Frost in [4] distinguish several classes of buffer sharing policies based on the time instances at which control actions can be taken and on the groups of cells that can be discarded. The three classes of policies studied here fall within that framework. Discarding type policies have been studied by Petr and Frost in [3, 5]. In [3] the problem of minimizing the average discarding cost has been considered in a system with two priority classes and one buffer space. In [5] the problem of optimizing the discarding cost under constraints on the losses of each class is considered. Here we consider a system with multiple traffic classes and buffers of arbitrary length and we optimize over *all* discarding policies.

The pushout scheme is another buffer sharing strategy that has been studied extensively in the past. An important component of a pushout strategy is to decide which cell to push out of the buffer in order to make space for an incoming cell. Kroner and Kroner *et al* have analyzed the performance of several pushout schemes in [1, 2] and obtained the cell loss probabilities of the different traffic classes. In our work we identify two important properties of the optimal pushout policy. It is better to push out the oldest low priority cell from the buffer and it is better to push out a low priority cell from the buffer in order to make space for another cell, irrespective of its priority. These two properties uniquely characterize the optimal pushout policy, called the squeeze-out policy, as we show in section 3.1. The class of expelling policies has been identified in [4] but they haven't been analyzed. Some related work in exponential queueing systems was done in the past by Lippman in [10].

The paper is organized as follows. In section 2 the discarding policies are described. The pushout and the expelling classes of policies are described in subsections 3.1 and 3.2 respectively. For the sake of brevity, proofs of theorems have not been presented here. The proofs are available in [11, 12, 13]. In section 4 we discuss some extensions to our work and open problems. In section 5 numerical results are reported.

2. Discarding Policies

The cells are classified into L priority classes. The high priority classes are more sensitive to cell losses. Without loss of generality we assume that the priority of class l is higher than the priority of class $l+1$. The priority of a class is reflected by the cost that is incurred by the blocking of a cell of that class. As we mentioned earlier at most B_T cells of all classes arrive into the system during every slot and they reside in the temporary buffer. By the end of each slot a decision is taken regarding which cells will be admitted in the system and where they are going to be placed in the buffer. The rest of the cells are discarded. We denote by $X_i^M(t)$ the class of the cell residing at the main buffer position i, $i = 1,..,B$ by the end of slot t; $X_i^M(t) = 0$ if position i is empty at this time. We denote by $X_i^T(t)$ the class of the cell residing at position i of the temporary buffer $i = 1,..,B_T$; $X_i^T(t) = 0$ if this position is empty at this time. The vectors $\mathbf{X}^M(t) = (X_i^M(t) : i = 1,..,B)$, $\mathbf{X}^T(t) = (X_i^T(t) : i = 1,..,B_T)$, represent the main and temporary buffer occupancies at the end of slot t. Without loss of generality we may assume that in the temporary buffer, the cells are stored in decreasing priority order and in contiguous buffer spaces; that is, for $X_i^T(t) > 0$, $i > 1$, we have $0 < X_{i-1}^T(t) \leq X_i^T(t)$. The temporary buffer at the end of slot t contains cells that arrived during slot t only. We assume independent identically distributed arrivals from slot to slot. The vector $\mathbf{X}(t) = (\mathbf{X}^M(t), \mathbf{X}^T(t))$ is a natural state variable and we use the notation $\{\mathbf{X}(t),\ t \geq 0\}$ for the stochastic process that describes the evolution of the system. The state space of that process is $\mathcal{X} = \mathcal{X}^M \times \mathcal{X}^T$ where $\mathcal{X}^M = \{0, 1, .., L\}^B$ and $\mathcal{X}^T = \{0, 1, .., L\}^{B_T}$ are the spaces where the vectors $\mathbf{X}^M(t)$ and $\mathbf{X}^T(t)$ lie respectively.

All the cells in the temporary buffer, by the end of each slot t, are either admitted in the system and placed in the main buffer or rejected. We control the admission of cells in the main buffer. The control actions taken by the end of slot t are represented by the *admission* variables $A_i(t) \in \{0, 1, .., B\}$, $i = 1, .., B_T$ as follows. We have $A_i(t) = 0$ if either position i of the temporary buffer is empty or the cell stored in that position is blocked from admission into the system; we have $A_i(t) = j$ if the cell residing in position i of the temporary buffer is placed in position j of the buffer. The vector $\mathbf{A}(t) = (A_i(t) : i = 1, .., B_T)$ is called the admission vector at time t in the following. Let $\mathcal{A} = \{0, .., B\}^{B_T}$ be the space where it lies; this is called the action space in the following. We assume that the cells of the temporary buffer which are admitted in the main buffer are placed in consecutive positions at the end of the existing queue. Let $S(\mathbf{x})$ be the set of all admission vectors which satisfy the above assumption when the system is in state $\mathbf{x}$.

At each slot t exactly one cell is transmitted. The cells in the main buffer are served in a FIFO manner.

An ***admission policy*** is any rule for selecting the admission variables at every time $t \geq 0$. This decision is made on the basis of the past system states $\{\mathbf{X}(s),\ t \geq s \geq 0\}$ and past decisions. Let G be the class of all admission policies such that the admission vector $\mathbf{A}(t)$ belongs to the set $S(\mathbf{X}(t))$ at all t.

When a cell of class l is dropped from the system then a cost c_l is incurred. We assume that the classes are indexed in decreasing priority, that is $c_l > c_{l+1}$, $l = 1, .., L-1$. By convention we set $c_0 = 0$. The total cost incurred when the system is in state $\mathbf{x}$ and the admission actions that correspond to vector $\mathbf{a} \in S(\mathbf{x})$ are taken is

$$c(\mathbf{x}, \mathbf{a}) \stackrel{\text{def}}{=} \sum_{i=1}^{B_T} 1\{a_i = 0\} c_{x_i^T}, \ \ \mathbf{x} \in \mathcal{X}, \ \mathbf{a} \in S(\mathbf{x}). \quad (1)$$

The blocking cost incurred at time t is $C(t) = c(\mathbf{X}(t), \mathbf{A}(t))$. Our objective is to minimize the average blocking cost. The long run average cost associated with a policy $g \in G$ is defined by

$$J_g(\mathbf{x}) \stackrel{\text{def}}{=} \limsup_{T \to \infty} E_{\mathbf{X}}^g \frac{1}{T} [\sum_{t=0}^{T-1} C(t)], \ \ \mathbf{x} \in \mathcal{X} \quad (2)$$

where $E_{\mathbf{y}}^g[\cdot]$ denotes the expectation with respect to the probability measure induced by the policy g on the state process starting in state $\mathbf{y}$. An admission policy g_D is said to be average cost optimal discarding policy if it minimizes (2) within G, i.e., if

$$J_{g_D}(\mathbf{x}) \leq J_g(\mathbf{x}), \ \ \mathbf{x} \in \mathcal{X}$$

for any other policy $g \in G$. Under our assumptions about the arrival statistics, the optimization problem associated with (2) falls within the family of discrete time Markov Decision Processes (MDP's). Since the state space is finite , it is well known that an optimal policy exists and it can be taken in the class of

Markov stationary policies [14]. The following theorem provides a structural characterization of the optimal policy.

Theorem 1: There exists an average cost optimal discarding policy g_D of the following form. There are thresholds $t_1 \geq t_2 \geq \cdots \geq t_L$ such that a cell of class j in position k of the temporary buffer is accepted if and only if

$$t_j \geq i + k$$

where i is the length of the main buffer.

The proof of the above theorem can be found in [11, 12, 13].

3. Cell expelling policies

In the following two sections we study the pushout and expelling class of policies. The main difference between these policies and the discarding policies is that cells in the main buffer can be dropped (expelled) from the system under the pushout and expelling policies. The pushout policies, where a cell can be expelled or discarded only if there is no space in the main buffer, constitute a subclass of the general expelling policies where cells can be expelled or discarded at any time.

We keep the notation that we introduced in section 2 in this section as well. Nevertheless we prefer to specify the class of policies we consider and the optimal policies in words rather than mathematically in this section, since the first description is precise enough and we don't need the mathematical description in the proof of the results.

3.1 Pushout policies

The class of pushout policies G^p contains all policies which obey the following rules.

a) A cell can be expelled from the main buffer only if it is "pushed out" by another cell in the temporary buffer which cannot enter the main buffer because it is full.

b) A cell from the temporary buffer can be discarded only if the main buffer is full.

The following policy is optimal in G^p.

Squeeze-out Policy π^{po}:

Append the cells from the temporary buffer to the end of the main buffer, high priority cells first.

If the main buffer is full, and there are cells in the temporary buffer, push out the low priority cells starting from those closest to the head of the queue.

If all the low priority cells in the main buffer are pushed out, and there are still cells in the temporary buffer, discard them.

The policy π^{po} is optimal within G^p in a very strong sense. It minimizes at every slot t the number of high priority cells lost. Furthermore this holds for arbitrary arrival processes and not only for i.i.d. arrivals. Let $D^h(t), D^l(t)$ and $\tilde{D}^h(t), \tilde{D}^l(t)$ be the numbers of dropped cells by the end of slot t of the high and low priority classes respectively under policy π^{po} and for an arbitrary policy $\tilde{\pi} \in G^p$. Then we have the following.

Theorem 2: When the system starts from the same initial state under policies π^{po} and $\tilde{\pi}$, and the arrivals are identical under the two policies, we have

$$D^l(t) \geq \tilde{D}^l(t)$$

$$D^l(t) + D^h(t) = \tilde{D}^l(t) + \tilde{D}^h(t) \quad t = 1, 2, \cdots$$

3.2 Expelling policies

The class of expelling policies G^E has as members all policies that append the new cells from the temporary buffer at the end of the queue and do not rearrange the cells in the main buffer. An expelling policy is allowed to expel or block any cell in the main or temporary buffer respectively, irrespective of the state. Hence the only requirement an expelling policy should satisfy is to preserve the FIFO order. Other than that it can drop cells arbitrarily. Clearly the class of expelling policies is wider than the previous two.

We were able to obtain properties of the optimal policy that narrow down the class of policies that contains the optimal policy significantly. We have shown that an optimal policy within G^E should act according to the following two rules.

1. The cells are placed from the temporary buffer to the main buffer, high priority cells first. If they do not fit then low priority cells are expelled starting from those closest to the head of the queue.

2a. If the cell at the head of the queue is of high priority then it is served.

2b. If the cell at the head of the queue is of low priority then either that cell is served, or all the low priority cells from the head of the queue until the high priority cell closest to the head of the queue are expelled, and that high priority cell is served.

Note that the two rules above characterize the optimal actions completely for some states and in general up to a binary decision of whether all low priority cells in the head of the queue are dropped or none of them. As in the case of pushout policies the above result holds for arbitrary arrival processes. Let G^{EO} be the class of policies which satisfy the above two rules. We claim that the optimal policy within G^{E} should belong to G^{EO}. More specifically we show the following.

Theorem 3: For every policy $\pi \in G^{E}$ there exists a policy $\tilde{\pi} \in G^{EO}$ such that if the system starts from the same initial state under the two policies and the arrival process is identical under the two policies we have

$$\tilde{D}^h(t) \leq D^h(t)$$
$$\tilde{D}^h(t) + \tilde{D}^l(t) \leq D^h(t) + D^l(t) \quad t = 1, 2, \cdots \qquad (3)$$

where $\tilde{D}^h, \tilde{D}^l(t)$ is the number of lost cells of high and low priority respectively under $\tilde{\pi}$ and similarly for $D^h(t), D^l(t)$ under π.

Note that in the theorem, $\tilde{D}^h(t) + \tilde{D}^l(t) \leq D^h(t) + D^l(t)$ implies $\tilde{D}^l(t) - D^l(t) \leq D^h(t) - \tilde{D}^h(t)$, which means even if $\tilde{D}^l(t) > D^l(t)$, the difference between them will be less than or equal to that between $D^h(t)$ and $\tilde{D}^h(t)$. Therefore, theorem 3 implies the discarding cost in $\tilde{\pi}$ will be less than or equal to that in π. The details of the proofs for the theorems in this section can be found in [12, 13].

4. Discussion and Open Problems

The problem of buffer management at an output link of an ATM node was considered in the paper. Three classes of policies were studied and optimal policies with respect to losses were identified. The classes of policies that have been considered are implementable by the architectures proposed in [7] using the Sequencer chip.

Regarding the assumptions about the arrivals, for the expelling and pushout policies our results hold for any arrival process while for the discarding policies *i.i.d.* arrivals were assumed. If Markov modulated arrivals are considered in the latter case, then during the periods at which the states of the underlying Markov processes of the arrivals are frozen the arrivals are *i.i.d.* and a threshold type of policy will be optimal. If the state of the underlying Markov process of the arrivals is included in the state description of the system together with the queue lengths then the optimal policy is conjectured to be of threshold type again but the thresholds will be functions of the underlying state as well.

We believe that policies analogous to the squeeze-out policy and the optimal expelling policy class can be used for the buffer control of packet-switched networks with variable sized packets and loss priorities. Two examples of such networks are Frame Relay and Packet Transfer Mode(PTM) networks[8]. We also believe that the results presented in this paper for pushout and expelling policies can be extended to a node modeled by an M/M/1/k queue fed by two classes of customers whose (exponential) service time distributions are identical.

In our study we focused on the performance degradation due to blocking. Another important performance measure is the delay experienced by the cells in the output link buffer. The issue of delay experienced by traffic streams multiplexed through a common transmission link has been studied extensively. An important open problem for further investigation is the joint consideration of loss and delay requirements. Scheduling policies which attempt to satisfy simultaneously certain delay and loss requirements need to be investigated. The ultimate goal remains to be the study of the buffer management control schemes as they interact at the network level in different nodes. This interaction determines the end-to-end system performance.

5. Numerical results

Figures 2 and 3 display some of the preliminary numerical results we have obtained. The objective was to compare the performance of some of the policies discussed in this paper. A two-priority system with i.i.d. arrivals was considered. The arrival process is derived from a binomial distribution and is the same as the one used in [5]. The arrival rate as well as the fraction of traffic from the two priority classes was varied. The cost of losing a high priority cell was varied from 10 to 10^8 times the cost of losing a low priority cell. Value iteration[14] was used to compute the performance of the optimal discarding, expelling and pushout policies as well as the default policy. The default policy is the one where cells are simply admitted to the main buffer in FIFO order, high priority cells first, and dropped if it is full. We considered a system with main and temporary buffer sizes of 7 and 3, respectively. The squeeze-out and default policies corresponded to single points in the plots in Figure 2 and Figure 3 since they are unaffected by the discarding costs. The performance of other pushout policies are provided for comparison. In both of these policies, low priority cells from the temporary buffer do not push out low priority cells from the main buffer but are dropped instead. In last-in-first-drop and first-in-first-drop(LIFD and FIFD) pushout policies high priority cells push out the low priority cells that are, respectively, furthest from and closest to the head of the queue. Note that for most cases considered, there is little difference in the performance of the LIFD pushout, FIFD pushout and squeeze-out policies. As expected, the expelling policy performed better than the discarding policy; the difference between the two policies depended on the total traffic and relative proportions of the two classes of traffic. For the optimum expelling policy, the decision of whether to serve the low priority cell at the head of the queue or to serve the high priority cell closest to the head of the queue was found to be almost completely dependent on the number of high priority cells in the main buffer. An expelling policy which made this decision based on a threshold on the number of high priority cells in the main buffer achieved results very close to the optimum expelling policy. This suboptimal policy could therefore be used as a basis for a practical implementation of the expelling policy.

References

[1] H. Kroner, "Comparative performance study of space priority mechanisms for ATM networks," *Proceedings of the IEEE INFOCOM Conference*, San Francisco, June 1990.

[2] H. Kroner, G. Hebuterne, P. Boyer, A. Gravey, "Priority management in ATM switching nodes," *IEEE Journal on Selected Areas in Communications*, 418-427, April 1991.

[3] D.W. Petr, V.S. Frost, "Optimal packet discarding: an ATM-oriented analysis model and initial results," *Proceedings of the IEEE INFOCOM Conference*, San Francisco, June 1990.

[4] D.W. Petr, V.S. Frost, "Priority cell discarding for overload control in BISDN/ATM networks: an analysis framework," *International Journal of Digital and Analog Communication Systems*, Vol. 3, No. 2, April-June 1990.

[5] D.W. Petr, V.S. Frost, "Nested threshold cell discarding for ATM overload control: optimization under cell loss constraints," *Proceedings of the IEEE INFOCOM Conference*, Florida, April 1991.

[6] A.W. Berger, A.E. Eckberg, T.C. Hou and D.M. Lucantoni, "Performance characterizations of traffic monitoring, and associated control mechanisms for broadband "packet" networks," *IEEE GLOBECOM*, pp. 350-354, 1990.

[7] H.J. Chao, N. Uzun, "A VLSI sequencer chip for ATM traffic shaper and queue manager," *IEEE Journal of Solid-State Circuits*, 1634-1643, November 1992.

[8] I. Gopal, R. Guerin, "Network Transparency: The plaNET Approach," *Proc. INFOCOM'92*, Florence, Italy, 1992.

[9] Y. Yeh, M.G. Hluchyj, A.S. Acampora, "The knockout switch: a simple, modular Architecture for high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, 1274-1283, October 1987.

[10] Steven A. Lippman, "Applying a new device in the optimization of exponential queuing systems," *Operations Research*, Vol. 23, No 4, July-August 1975.

[11] L. Tassiulas, Y. Hung, S. Panwar, "Optimal cell admission control in an ATM network node," *Proceedings of the 1992 Conference on Information Sciences and Systems*, Princeton, NJ, March 1992.

[12] L. Tassiulas, Y. Hung, S. Panwar, "Optimal buffer control during congestion in an ATM network node," submitted to the *IEEE Transactions on Networking*.

[13] Y. Hung, "Buffer Management Policies for Broadband Networks," Ph.D. Dissertation, Department of Electrical Engineering, Polytechnic University, Brooklyn, NY, Jan. 1993.

[14] S. Ross, *Introduction to dynamic programming*, Academic Press, New York,NY, 1983.

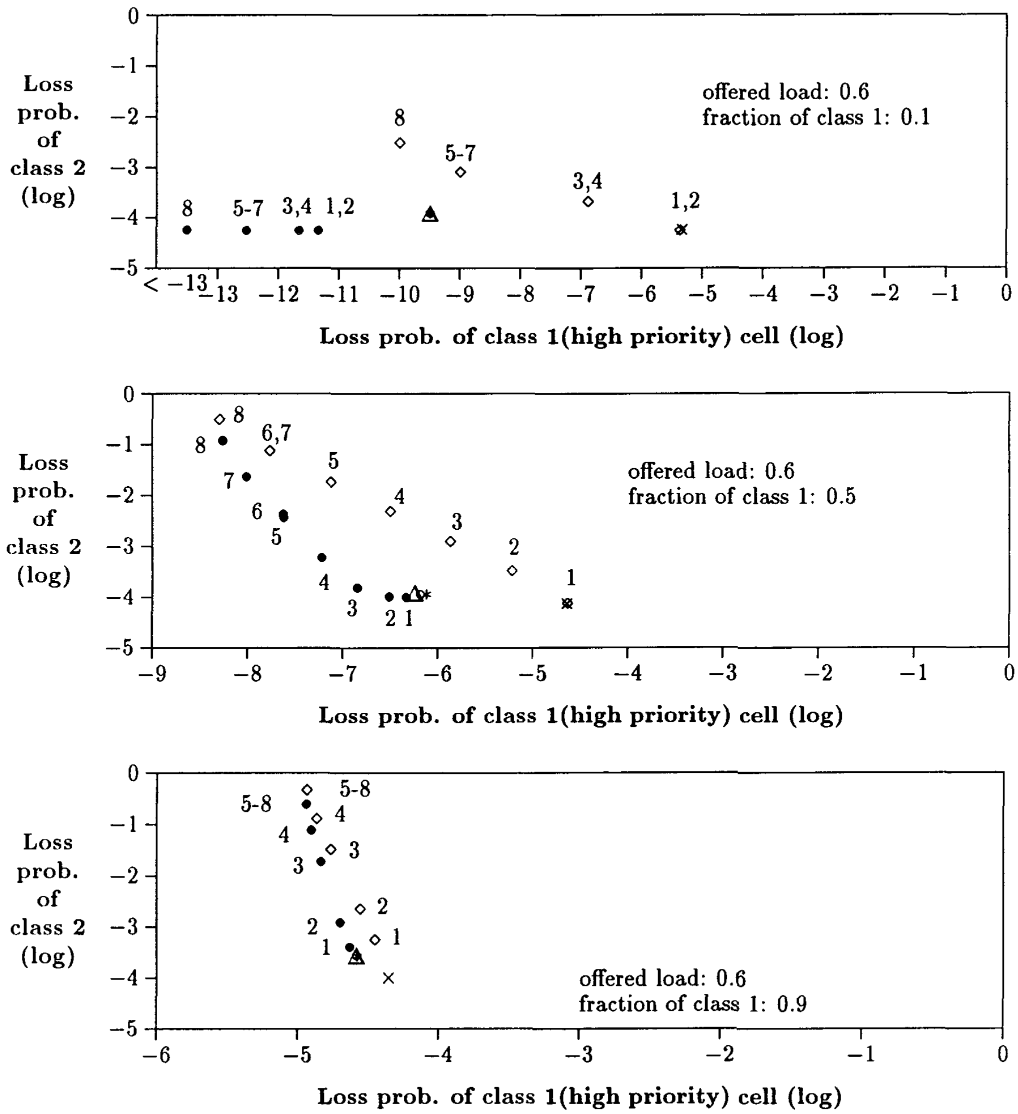
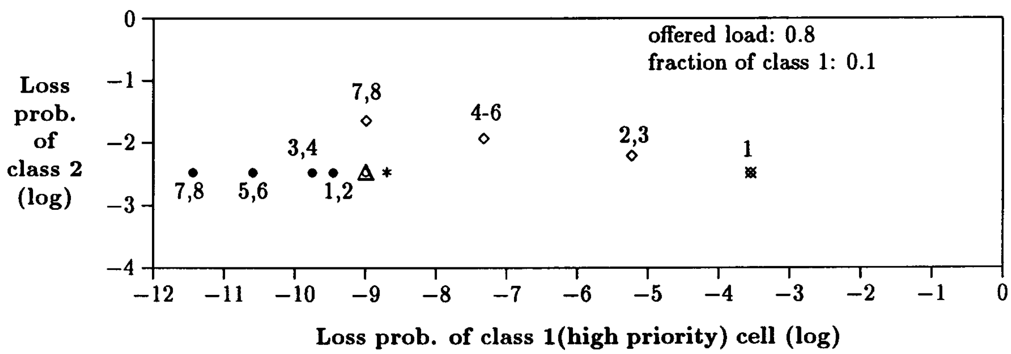
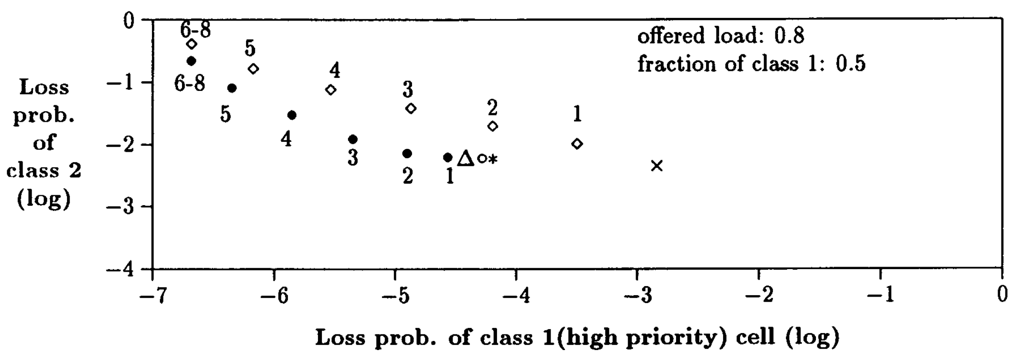


Figure 2: Loss probabilities for two classes with main buffer size=7, temporary buffer size=3, discarding cost of a low priority cell=1, discarding cost of a high priority cell changes from 10 to 10^8. In the figure, "Δ" stands for squeeze-out policy, "*" for LIFD pushout, "∘" for FIFD pushout, "◇" for discarding policy, "×" for default policy, which sets the thresholds of both classes to be the main buffer size, and "•" for expelling policy. The numbers next to the "◇" and the "•" stand for the powers, of 10, of the discarding costs of class 1 cells used to obtain those points.





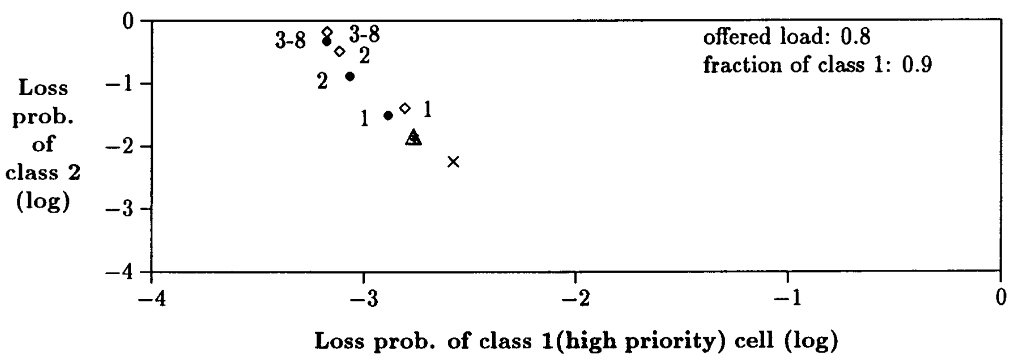


Figure 3: Loss probabilities for two classes with main buffer size=7, temporary buffer size=3, discarding cost of a low priority cell=1, discarding cost of a high priority cell changes from 10 to 10^8. In the figure, "Δ" stands for squeeze-out policy, "*" for LIFD pushout, "∘" for FIFD pushout, "⋄" for discarding policy, "×" for default policy, which sets the thresholds of both classes to be the main buffer size, and "•" for expelling policy. The numbers next to the "⋄" and the "•" stand for the powers, of 10, of the discarding costs of class 1 cells used to obtain those points.