On Protective Buffer Policies

Israel Cidon, Roch Guérin, and Asad Khamisy

Abstract—**We study buffering policies which provide different loss priorities to packets/cells, while preserving packet ordering (space priority disciplines). These policies are motivated by the possible presence, within the same connection, of packets with different loss probability requirements or guarantees, e.g., voice and video coders or rate control mechanisms. The main contribution of the paper is the identification and evaluation of buffering policies which preserve packet ordering and guarantee high priority packets performance (loss probability), *irrespective* of the traffic intensity and arrival patterns of low priority packets. Such policies are termed *protective policies*. The need for such policies arises from the difficulty to accurately characterize and size low priority traffic, which can generate large and unpredictable traffic variations over short periods of time. We review previously proposed buffer admission policies and determine if they satisfy such "protection" requirements. Furthermore, we also identify and design new policies, which for a given level of protection maximize low priority throughput.**

Keywords—**ATM, Priority, Buffers**

1 Introduction

This paper deals with the study of buffering policies that arise in the context of fast packet-switched networks carrying different classes of traffic [1, 2]. These networks are faced with the difficult task of satisfying the needs of connections requiring different Qualities Of Service (QOS), but sharing the same physical resources, e.g., bandwidth and buffers. Policies providing different performance levels to several traffic classes are a subject of great current interest as evident from the literature and standard proposals.

Most policies fall in either one of two categories. Service scheduling policies which implement different delay classes by arranging the order in which packets are served, and buffer admission policies which enforce different loss probability classes by selectively discarding packets. The first are often referred to as *time priority* disciplines [3], and differentiate between packets from separate sessions with different service requirements. The second, termed *space priority* disciplines [4, 5], discriminate between packets without changing their ordering. Hence, they can also be used within the same session. Policies combining both aspects are discussed in [6].

The paper is concerned with space priority disciplines which enforce different priority classes within the same session. Specifically, we consider a buffer shared by high and low priority packets, where the buffering policy enforces different loss probabilities for the two classes while preserving packet ordering. Such policies are motivated by the possible presence within the same session of packets with different loss probability requirements. The selective discarding (in case of congestion) of lower priority packets must, therefore, be done while maintaining in sequence delivery. Our focus is on identifying buffering policies that can provide performance (loss) guarantees, to high priority packets, *irrespective* of the arrival intensity and patterns of low priority packets. We identify and evaluate such policies, while assessing the cost (loss probability) of this protection to low priority packets.

There are many examples of connections carrying packets with different loss priorities. For example, voice or video coders designed to take advantage of statistical multiplexing [7, 8] split the information they generate into its most and least significant components, which are transmitted in packets of different loss priorities. Bandwidth is then allocated to sustain the reliable transmission of high priority packets and deliver low priority ones only when network conditions permit it. As the aggregate volume of high and low priority packets can often exceed the allocated bandwidth, low priority packets may be discarded. This remain acceptable as these packets represent enhancements to the information carried in high priority packets. The more important aspect is to ensure that the loss probability of high priority packets is not adversely affected by these potential load increases.

Connections with packets belonging to different priority classes are also created by the *marking* feature of the mechanisms used to regulate the flow of packets into the network [1, 9]. For each connection, packets in excess of the "acceptable" flow are either discarded before entering the network, or sent but tagged as *excess* or low priority traffic. The amount of excess traffic that each connection is allowed to generate can be properly controlled, but it is difficult to control the aggregate volume of excess traffic offered at any time to each network link. It is, therefore, important to protect "regular" packets from potential low priority traffic surges.

A number of space priority policies have been proposed and analyzed, in the literature, e.g., [10, 4]. The emphasis

Israel Cidon is now with Sun Microsystems, 2550 Garcia Avenue, Mountain View, CA 94043-1100. He is also with the Department of Electrical Engineering, Technion, Haifa 32000, Israel. Part of this work was done while he was the IBM T.J. Watson Research Center. (email: israel.cidon@Eng.Sun.Com)

Roch Guérin is with the IBM T.J. Watson Research Center Yorktown Heights, NY 10598. (email: guerin@watson.ibm.com)

Asad Khamisy is with the Department of Electrical Engineering, Technion, Haifa 32000, Israel. Part of the work was done while visiting the IBM T.J. Watson Research Center. (email: asad@ts.technion.ac.il)

of these works has, however, mainly been on determining acceptable load regions as a function of the loss requirements of each priority class. This paper differs from these earlier works as it focuses on understanding and defining when and under which assumptions, performance guarantees hold for high priority packets. Specifically, we wish to define policies which guarantee loss probability to high priority packets with no assumptions on the arrival patterns of low priority ones. We term such policies *protective,* and as mentioned earlier we feel that the investigation of such a property is justified by the difficulty of characterizing or even sizing low priority traffic, which is often subject to large and unpredictable variations. This clearly does not mean that one should ignore the loss probability offered to low priority packets once the high priority ones are protected. Policies that do well in this respect are also investigated in this paper.

Our goals are, therefore, to first define the performance guarantees that protective policies should provide to high priority packets, and then identify policies which meet these requirements. In particular, it is of interest to determine which, if any, of previously proposed buffer admission policies can provide such guarantees. In addition, we also compare policies on the basis of how well they fare with low priority packets, and devise policies that attempt to maximize low priority throughput while remaining protective.

The organization of the paper reflects the above goals. In section 2, we define more formally the system and the performance guarantees to be provided to high priority packets. In section 3, we review a number of buffer admission policies that have been previously proposed, and determine which satisfy the "protection" requirements set forth in the previous section. In section 4, we define several new protective policies, with the goal of improving the handling of low priority packets. Section 5 gives numerical examples that illustrate the findings of the paper, while a brief summary of the main results is provided in section 6.

2 Model

The input to the *system* is a cell stream formed from the superposition of two cell sources, the *green* and the *red* sources. The cell generation process of both sources (in particular the red source) is arbitrary and typically unpredictable. The green cells source represents the superposition of all "high priority" cells generated by all users sharing the link, and the red cells source represents the superposition of all "low priority" cells generated by these users. Both sources share a buffer of size N cells, which is served by a fixed rate server. A slotted system representative of an ATM environment is assumed (continuous time systems are considered in [11]). A slot corresponds to the transmission time of a cell, cell transmissions are aligned with slot boundaries and buffers are released after transmission is completed. We further assume that some *policy* is employed to decide whether to accommodate/discard an arriving cell and drop/serve a stored one.

An M *protective* policy (where $N = M + K$, and $K \geq 0$) is defined as follows. Assume that each cell generated by the *green* source is duplicated, with this duplicate stream fed into another finite queue with M cell buffers and a similar fixed rate server. This latter system, which is restricted to green cells, is called the *reference system.* The former (that accommodates both colors) is termed the *main system* or simply the system. The policy used in the reference system is to accommodate a green cell if a buffer is available (i.e., drop it only if all buffers are occupied), and to serve the cells according to a FCFS ordering. A policy is M *protective* in the *strong sense* (see below), if for any sequence of arrivals of red and green cells, the green cells lost in the system are a subset of the green cells lost in the reference system. An M protective policy guarantees green cells a service better than or equal to that of a system with M buffers dedicated to green cells only.

The above definition is termed *strong* because of its subset requirement. In practice it may be sufficient to guarantee that the *number* of green cells lost in the system is less than or equal to the number of (green) cells lost in the reference system. We refer to such policies as *protective* in the *weak sense.* Policies that are protective in the strong sense have the advantage of also ensuring that loss patterns are not significantly altered by the presence of red cells, while policies which are protective in the weak sense can only make guarantees with respect to the number of lost cells. One scenario which may help illustrate the potential significance of such a difference is that of system where the number of lost green cells are identical in both systems, but are grouped together in the reference system and spread out evenly in the main system. The latter may then correspond to a much larger number of lost user frames, i.e., all lost cells are from different frames, which translates into more severe performance degradations for the user. Fortunately, all the discrete time policies we investigate satisfy either both definitions or neither. In the rest of the paper we further narrow our focus to policies which are *FCFS M protective,* i.e., M *protective* and preserving cell ordering.

3 Existing Buffer Policies and Their Properties

In this section, we consider three well known policies which discriminate between low and high priority cells, and examine their ability to satisfy the above protective definition. This section has two purposes. First, it shows that three policies which appear reasonably similar in terms of performance, actually offer different levels of "protection." Second, it provides a better intuitive understanding of the key characteristics of protective policies, which helps design better protective policies.

3.1 The Pushout Policy

The pushout policy attempts to protect green cells by discarding red ones first in case of congestion (a full queue).

If an empty buffer is available, the pushout policy always accommodates a cell irrespective of its color. If no buffer is available the action depends on the cell color. If it is red, it is discarded. If it is green and red cells are currently buffered, one of them is pushed out and the green cell is accommodated at the end of the queue. Otherwise, the arriving green cell is discarded. Pushout policies can be further classified as a function of the rule used to determine which red cell to push out, e.g., first, last, random. The pushout policy can also be generalized by allowing a red cell arriving to a full queue to push out another already buffered red cell. Our analysis of the protective properties of the pushout policy applies to all such variations.

It is clear that the pushout policy is FCFS. Traditionally, it had been considered as complex to implement, but as the best choice from a performance point of view [4], especially in terms of red throughput for a desired level of performance to green cells. The intuition is that the pushout policy utilizes available buffers as well as possible by refraining from discarding red cells as long as no congestion is present. The following proposition, however, states that it is quite limited in protecting green cells.

Proposition 1 *The pushout policy is not M protective for all $M > 1$ and all $N \geq 1$.*

Proof: The proof is trivial if $N < M$, therefore, we assume that $N \geq M$. Since by definition a policy must be protective for any sample path, we concentrate on a specific scenario. (Later, we illustrate via numerical examples that the "lack of protection" extends to more general cases.)

Consider the following arrival pattern: At $t = 0$, N red cells are generated. Then, at $t = 1, 2, \ldots, N$ a single green cell is generated and no red cells. At $t = N + 1$, a batch of M green cells is generated (and no red cells). It is clear that for all times $t = 1, 2, \ldots, N$ the number of cells in the reference system is exactly one (green) and it is N in the main system. Furthermore, as the cells are served according to the FCFS policy, the N cells queued in the main system at time $t = j$ consists of $N - j$ red cells followed by j green cells. Therefore, at time $N + 1^-$ there are no cells in the reference system and $N - 1$ green ones in the main system. Therefore, the batch of M green cells arriving at time $N + 1$ will be fully accommodated in the reference system but $M - 1$ green cells will be discarded in the main system. This implies that for any $M > 1$, the pushout policy is not M protective.

Repeating the above pattern periodically yields an infinite length sample path, for which the reference system is lossless and the main system loses $\frac{M-1}{N+M}$ of the green cells. Note that the result holds for both the strong and weak definitions of the protective property. ∎

3.2 The Limited Red Policy

The limited red policy was suggested in [10] as another policy for protecting green traffic from excessive red traffic, while providing an acceptable level of service to red cells. This policy allows the concurrent buffering of up to L red cells in the system. In other words, an arriving red cell is accepted if and only if there is a free buffer available and less than L red cells are currently in the queue. Green cells are accommodated if there is a free buffer. This policy avoids discarding already buffered cells, and only requires that the number of buffered red cells be tracked. This results in a lower implementation complexity than the pushout policy. We show next that the limited red policy, like the pushout policy, does not satisfy either of the protective definitions.

Proposition 2 *The limited red policy is not M protective for all $M > 1$, $N \geq 1$ and $L \geq 1$.*

Proof: The proof is straight forward if $N < M + L$. Therefore, we assume that $N \geq M + L$ and define k as the largest integer such that $kL \leq N$. As for the pushout policy, the proof consists of exhibiting a specific sample path for which the protective property is violated. This sample path is provided here (we focus on the main system) for the simple case where $k = 2$ and $N = 2L$ ($M \leq L$). Its extension to general values of k and N is straightforward and can be found in [11].

At $t = 1$, L red cells are generated together with a single green cell. Next, one green cell is generated in each subsequent slot, so that the total number of cells in the system remains constant. This is continued until the initial L red cells have all been replaced by green cells, i.e., until time $t = L$. Therefore, just before time $L + 1$ the main system stores exactly L green cells, while the reference system only contains a single (green) cell as it sees a single arrival per slot. At this point, another batch of L red cells is generated, and all are again accommodated. The buffer in the main system is then full with $2L$ cells, of which the first L are green and the last L are red. In order to replace all cells in the buffer by green cells, a single green cell is generated in each of the next $2L$ slots. At the end of this period, the buffer of the main system contains $2L$ green cells, while the reference system only stores a single green cell. At this point, a batch of M green cells is generated and all are lost in the main system, while they are all accommodated in the reference system. The above arrivals pattern can be periodically repeated so that the main system loses ($\frac{M}{k(k+1)L/2+N+M}$) of the green cells (while the reference system is lossless). ∎

3.3 The Threshold Policy

The threshold policy is another popular policy that discriminates between green and red cells. It accommodates red cells, if and only if the number of occupied buffers (by cells of either color) is less than some integer T. Its main advantage is a low implementation complexity as only the total number of buffered cells needs to be tracked. If $N < M + T$ it is easy to show that the policy is not M protective. Therefore, we focus on the case $N \geq M + T$ for which we show the following.

Proposition 3 *The threshold policy is M protective for all $T \geq 0$ and all $N \geq M + T$.*

Before proceeding with the proof, we introduce a simple *invariant* which relates the number of buffers available to green cells in the main system to that of the reference system. A system that possesses this invariant is protective.

Lemma 1 *A policy is protective (in the strong sense) if and only if:*
(a) At all times the number of buffers available to green cells in the main system is at least equal to that of the reference system;
(b) If the reference system accommodates a green cell, then so does the main system.

Clearly, properties (a) and (b) together ensure the protective property and the converse is also true. The main aspect of the lemma is the invariant (a) which captures a key behavior of the protective system. Whenever it is violated an appropriate burst of green cells will result in the loss of the protective property. Our counter examples for the pushout and limited policies eventually resulted in fewer buffers available to green cells in the main system despite the greater initial number of buffers. As we show next, the threshold policy avoids such scenarios by discarding red cells early on at the onset of congestion.

Proof of proposition 3: The proposition is proved by showing that the threshold policy maintains the invariants of lemma 1. We prove this by using an induction on the sequence of events, that can affect either conditions of the lemma (the induction hypothesis). Therefore, the sequence includes only arrivals and departures. We assume $T > 0$ as the proof is trivial for $T = 0$.

The first event e_1, or starting point of the induction, must be an arrival and both the main and reference systems are empty at that time. Since $N > M$, the induction hypothesis remains true after e_1. Next, we assume that the hypothesis holds for all events e_i, $1 \leq i \leq n$, and proceed to show that it holds for $i = n + 1$.

Assume e_{n+1} is an arrival. If it is a green arrival and both systems have a free buffer, then the hypothesis still holds as both systems free buffer pools are reduced by one and both accommodate one additional green cell. If this green cell is accommodated only in the main system (property (b) is not violated), this implies that the reference system is full and thus has no free buffers. Therefore, invariant (a) still holds after this green arrival, and the induction hypothesis remains true. Note that the arriving green cell cannot be accommodated only in the reference system as the induction hypothesis holds prior to that arrival. If e_{n+1} is a red cell arrival, two cases must be distinguished. If the red cell is discarded in the main system, then nothing changes and the induction hypothesis remains true. If the red cell is accepted in the main system, then according to the threshold policy less than T buffers are currently occupied. Hence, after accepting the new red cell, the main system still has at least M free buffers. This guarantees that invariant (a) still holds, which ensures that the induction hypothesis remains true.

Next, assume that e_{n+1} is a departure. If both systems have a departure, then the hypothesis continues to hold. Otherwise, assume first that only the reference system has a departure (it must be a green departure). This requires that the main system be empty at the beginning of the corresponding service slot, which readily implies that the induction hypothesis still holds. If the departure is only from the main system, then the invariant clearly holds after that departure. This completes the proof that the threshold policy is M protective (in the strong sense). ∎

4 Improved Protective Policies

The previous section demonstrated the existence of protective policies and showed, that among the three policies considered only the threshold policy is protective. An open question is then to define "better" protective policies. An intuitive goodness criterion is the level of performance (cell loss) provided to red cells. The identification of the "best" protective policy is essentially an optimization problem under constraint, where we wish to minimize the loss of red cells while maintaining the protective property. It is an open and interesting question whether there exists a protective policy optimal (minimal loss of red cells) for arbitrary arrival patterns of red and green cells. However, it is possible to identify policies that improve upon the threshold policy. In this section, we first present a simple improvement to the threshold policy, using the intuition gained in the previous section. Next we show how greater improvements can be obtained by defining a new class of policies which further exploit the notion of invariant introduced earlier.

4.1 Extended Threshold Policy

A simple extension to the threshold policy comes from the realization that the loss of the protective property is caused by allowing red cells to excessively delay the progress of green cells. The threshold policy limits this by restricting red cells to the first T positions in the queue. This can be relaxed while ensuring that red cells still cannot delay the progress of green cells in the last M buffers by adding a limited pushout capability to the threshold policy. Red cells are now accepted whenever there is a free buffer, including above the threshold, but any red cell currently stored above the threshold is discarded upon the arrival of a green cell. This improves the performance seen by red cells as they can gain access to the last M buffers, while ensuring that they still do not delay green cells stored in these last M buffers by remaining "behind" any such cell. As we shall see in section 5, this does improve the loss probability of red cells when the green cell load is not too high, and it is easy to see that this Extended Threshold Policy (ETP) remains protective.

4.2 Policies Based on Reference System Simulation

Next, we introduce a class of protective policies which enhance the service of red cells by limiting the service of green cells to the minimum required to remain protective and by discarding only the red cells that cannot be served without violating the protective property. The main issue is to define criteria to recognize such cells. The approach we propose has two main components. The first consists of a set of properties of the reference system that should be preserved in the main system, and from which we can determine which cells to discard. The second component covers the method we rely on to preserve these properties.

A first property of the main system that we would like to preserve is the level of performance offered to green cells. Specifically, while we want to "protect" green cells, we also do not want to do more than that, i.e., provide them with a performance level above that of the reference system. Another important property of the main system is captured in the invariant of lemma 1 regarding the number of buffers available to green cells in both systems. The idea here is to ensure that the main system never falls behind the reference system in terms of this available buffer space. Note that we may consider buffers occupied by red cells as being available to green cells, i.e., red cells can be pushed out.

The approach used to guarantee the preservation of these two properties in the main system is based on *tracking* the behavior of the reference system by simulating it. This may appear like a complex task, but it can be easily implemented using for example similar components as for the leaky bucket algorithm [9]. The information available from this simulation allows us to (a) identify green cells which are not accommodated in the reference system, so they can be discarded in the main system; and (b) verify that the number of buffers available to green cells in the main system is always greater than or equal to the corresponding number in the reference system (invariant). In the rest of this section we describe protective policies, that rely on such a simulation and use it to provide increasingly better levels of service to red cells.

The first policy, the *simulated protective policy* or SPP, uses the information available from the simulated reference in a straightforward way to identify which green cells to accommodate and ensure that the invariant is never violated. The rules that determine the operation of SPP are as follows:

1. A red cell is always accommodated if a free buffer is available. *(As the red cell can be pushed out, the buffer remains available to green cells and the invariant cannot be violated.)*
2. A green cell is accommodated into a free buffer or pushes out a red one (assuming the buffer is full and red cells are present), if and only if it is accommodated in the reference system simulated in parallel. *(Note that SPP doesn't specify which red cell to push out, and thus a class of SPP policies can be defined in that fashion.)*
3. Before bringing a red cell into service it is checked whether the invariant still holds at the completion of its service time. If not, this and subsequent red cells are dropped until a green cell is encountered, which is then the one brought into service. *(The serving of a red cell may result in a violation of the invariant if the main and reference systems both contain the same (non-zero) number of available buffers at that time.)*

SPP is easily shown to be FCFS protective and is described more formally in [11]. Section 5 demonstrates its performance improvement over that of the threshold based policies. It is, however, possible to further improve SPP. The key to this improvement is to identify as early as possible red cells, that are bound to be discarded as their service will violate the invariant. SPP checks this only when the cell is about to enter service. Detecting and discarding these cells earlier would free the associated buffer and make it available to new arriving red cells that might otherwise have been dropped.

The following example further clarifies the proposed extension to SPP. Assume that the main and reference systems have 5 and 4 buffers, respectively, and that at time t there are two green cells in the reference system (the first one just entered service) while there are four cells in the main system. The order and colors of cells in the main system are as follows: The first cell is green and just entered service, the second is red, and the last two are green. The number of buffers available to green cells is two in both systems. At $t+1$, green cells complete service in both systems, and assuming that no arrivals occur between t and $t+1$, the reference system now contains a single green cell while the main system contains a red cell followed by two green cells. This red cell, however, will not be brought into service as doing so would result in violation of the invariant (at the end of its service the reference system would have four buffers available to green cells, while there would be only three in the main system). This conclusion could already have been reached at time t, so that the buffer could have been freed at that time.

The Extended SPP (ESPP) is a policy that capitalizes upon the above observation, and attempts to improve upon SPP by identifying as early as possible red cells which will be discarded before being brought into service. This is achieved by performing an additional check whenever a new green cell is accommodated. This procedure determines (see below) if "doomed" red cells exist in the main system at that time. The red cells are checked in their order of arrival, a doomed red cell is immediately discarded and the next red cell is checked if one exists in the buffer. This procedure ends either when all red cells in the buffer are discarded, or upon finding one which is not doomed. Note that the checked red cell is then preceded only by green cells as all its red predecessors (if any) have been discarded.

The determination of a doomed red cell proceeds as follows: Let G_p be the number of green cells ahead of the currently tested red cell, i.e., G_p slots must elapse before this red cell is brought into service. The number of buffers either empty or occupied by red cells in the remainder of the queue is then added to this value G_p. This number, denoted n_{test}, gives the number of buffers that will be available to green cells (assuming no new green arrival) when that red cell is about to enter service. Next, the number G_r of green cells in the reference system is determined and used to obtain the number $\min(M - G_r + G_p, M)$ of buffers that will be available to green cells in the reference system when the checked red cell is about to enter service (no green arrivals are again assumed). From this information, we conclude that the checked red cell is doomed and can be discarded if and only if $n_{test} < \min(M - G_r + G_p, M)$. Any red cell following a doomed red cell (with no green cell in between) can also be readily discarded as the value n_{test} remains unchanged.

This completes the description of the additional test performed for ESPP, and the performance improvements it yields are again illustrated in the next section. It should also be clear from the above discussion, that ESPP remains protective as it simply drops red cells earlier than SPP.

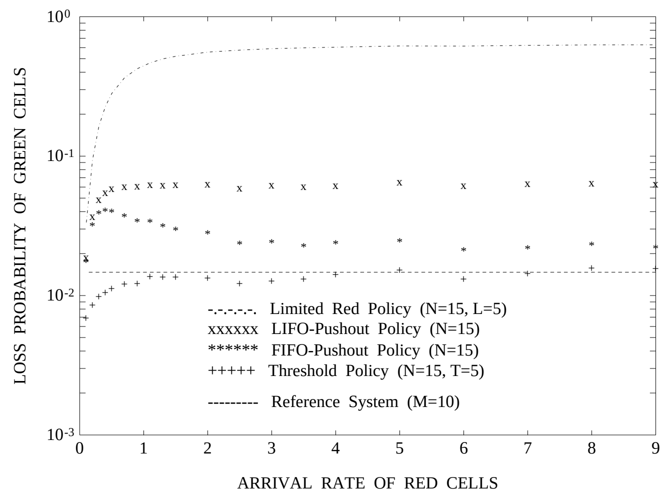

Figure 1: Loss probability of green cells as a function of λ_r for $N = 15$, $M = 10$ and $\lambda_g = 0.9$.

5 Numerical Results

In this section, we provide numerical results that quantify the concept of protective policy, and help compare the different policies that were discussed. We use a simple discrete time model (for a continuous time system see [11]), where the arrival process of green (red) cells is i.i.d. from slot to slot, and has a Poisson distribution with rate λ_g (λ_r). While analytical results are available for few cases [4, 10], analysis is not the focus of this paper and we use simulations to obtain loss probabilities for green and red cells under different policies. All simulations runs where over 1,000,000 slots long, and the same random generator seed was used for all cases. This means that for the same arrival parameters, identical sequences of cell arrivals are generated for all policies. This helps compare different policies as even if confidence intervals for the loss probability are large, relative differences remain meaningful.

The first set of results focuses on the differences between protective and non-protective policies. It shows that they are not limited to obscure sample paths tailored for counter examples, but are also found in common scenarios. Figure 1 gives for the three policies of section 3 with $N = 15$ and $M = 10$, the loss probability of green cells as a function of λ_r. In all cases, the value of λ_g was set to 0.9. For the pushout policy two replacement strategies were used: FIFO- and LIFO-pushout [4]. The loss probability of green cells in the reference system was included for comparison purposes and found equal to 0.0147.

Figure 1 shows, that under the threshold policy the loss probability of green cells does remain below this value irrespective of the red traffic intensity. On the other hand, under non-protective policies as the red load increases and despite the larger buffers, the green loss probability eventually exceeds this value. Figure 1 also points to the interesting fact, that among non-protective policies the FIFO-pushout gives the smallest loss probability for all values of λ_r. Moreover, its loss probability for green cells *decreases* as the arrival rate of red cells *increases* beyond 0.5. The latter can be intuitively explained by the fact, that increasing the arrival rate of red cells eventually increases the probability that an arriving green cell pushes out a red cell instead of joining a free buffer. In the case of FIFO-pushout, the earliest red cell (the closest to the server) is pushed out and, therefore, prevented from getting served. This reduces the waiting times of subsequent green cells and decreases their number in the system in the next slots. This in turn lowers their loss probability.

Next, we focus on red traffic performance and investigate the relative improvements offered by the policies of section 4. The comparison includes the threshold policy (TP), the extended threshold policy (ETP), the simulated protective policy (SPP) and the extended SPP with two strategies for red cells pushout; LIFO- (ESPPL) and FIFO (ESPPF). Table 1 gives for a system with $N = 20$, $M = 10$, the red cell loss probability for a range of cell arrival rates. While all tested policies are protective, the results show how they differ in the level of performance they give to red cells. The loss probability of red cells decreases as we move from the threshold policy (at the top of the table) to the ESPPF (at the bottom of the table). Note that the simulated protective policies (SPP, ESPPL and ESPPF), indeed decrease the loss probability of red cells. For example, for $\lambda_g = 0.5$ and $\lambda_r = 0.4$, the loss probability of red cells is about 5 times larger under the threshold policies than under the simulated protective policies.

6 Conclusions

In this paper, we have introduced the concept of *protective buffer policies* which guarantee loss probability to high priority cells and preserve ordering among cells, *irrespective*

POLICY	$\lambda_g = 0.5$			$\lambda_g = 0.8$		
	$\lambda_r = 0.4$	$\lambda_r = 0.5$	$\lambda_r = 0.6$	$\lambda_r = 0.1$	$\lambda_r = 0.2$	$\lambda_r = 0.3$
TP	0.0287944	0.0919565	0.1896974	0.0693228	0.1999560	0.3655125
ETP	0.0214789	0.0820025	0.1831950	0.0541368	0.1828037	0.3565503
SPP	0.0063104	0.0538020	0.1701969	0.0136643	0.1353798	0.3362013
ESPPL	0.0061379	0.0533402	0.1700031	0.0136443	0.1350749	0.3359099
ESPPF	0.0060604	0.0526725	0.1696154	0.0136443	0.1348700	0.3357458

Table 1: Probability of red cell loss with $M = 10$ and $N = 20$.

of the traffic intensity and arrival patterns of low priority cells. We first investigated well known policies, identified which were protective, and quantified these differences for some examples. We then developed new policies whose goal was to provide the best possible level of performance to low priority cells, while remaining protective. The ideas behind these policies were the concept of simulating in parallel a reference system, and using the information it provides to preserve key invariants and properties. Numerical examples showed, that these new policies were indeed successful at improving the performance of low priority traffic. We believe the notion of protective policy is important, especially when it is difficult to control the rate and patterns of low priority traffic.

Acknowledgement: The authors would like to thank Abhay Parekh for useful comments that helped clarify the presentation of several of the concepts presented in the paper.

References

[1] UNI specification draft 2.4. Technical report, ATM Forum: Technical Committee, August 1993.

[2] I. Cidon, I. Gopal, P. M. Gopal, R. Guérin, J. Janniello, and M. Kaplan. The plaNET/ORBIT high speed network. *Journal of High Speed Networks*, 2(3):1–38, September 1993.

[3] N. K. Jaiswal. *Priority Queues*. Academic Press, New York, 1968.

[4] H. Kröner, G. Hébuterne, P. Boyer, and A. Gravey. Priority management in ATM switching nodes. *IEEE Trans. Commun.*, COM-9(3):418–427, April 1991. (See also *Proc. INFOCOM'90.*).

[5] D. W. Petr and V. S. Frost. Nested threshold cell discarding for ATM overload control. In *Proc. INFOCOM'91*, pages 1403–1412, Boca Raton, Florida, 1991.

[6] A. Gravey and H. Hébuterne. Mixing time and loss priorities in a single server queue. In *Proc. ITC-13*, pages 147–152, Copenhagen, Denmark, June 1990.

[7] G. Karlsson and M. Vetterli. Packet video and its integration into the network architecture. *IEEE J. Select. Areas Commun.*, 7(5):739–751, June 1989.

[8] P. A. Probst and M. Yamashita. Coding schemes for voice and video, and their systems. *Telecommunication Journal*, 58(12):898–906, December 1991.

[9] I. Cidon, I. Gopal, and R. Guérin. Bandwidth management and congestion control in plaNET. *IEEE Commun. Mag.*, 29(10):54–63, October 1991.

[10] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high-speed packet switched networks. In *Proc. INFOCOM'90*, San Francisco, 1990.

[11] I. Cidon, R. Guérin, and A. Khamisy. On protective buffer policies. Research Report RC 18113, IBM Research Division, T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, June 1992.

Israel Cidon Israel Cidon (M'85-SM'90) received the B.Sc. (summa cum laude) and the D.Sc. degrees from the Technion - Israel Institute of Technology in 1980 and 1984, respectively, both in electrical engineering. From 1984 to 1985 he was on the faculty with the Electrical Engineering Department at the Technion. In 1985 he joined the IBM T.J. Watson Research Center, NY, where he was a Research Staff Member and a manager of the Network Architectures and Algorithms group involved in various broadband networking projects. Recently he joined Sun Microsystems Labs in Mountains View, CA, to initiate research activities in the area of high-speed networking. Since 1990 he is also with the Department of Electrical Engineering at the Technion.

He currently serves as an editor for the IEEE/ACM Transactions on Networking. Previously he served as the Editor for Network Algorithms for the IEEE Transactions on Communications and as a guest editor for Algorithmica. In 1989 and 1993 he received the IBM Outstanding Innovation Award for his work on the PARIS high speed network and topology update algorithms respectively. His research interests are high-speed wide-area and local networks, distributed algorithms and wireless networks.

Roch Guérin (S'84, M'86, SM'91) received the "Diplôme d'Ingénieur" from ENST, Paris, France, in 1983, and his M.S. and Ph.D. from the California Institute of Technology, both in Electrical Engineering, in 1984 and 1986 respectively. Since August 1986 he has been with IBM at the Thomas J. Watson Research Center, Yorktown Heights, New York, where he now manages the Network System Design group in the Advanced Networking Laboratory department. His current research interests are in the areas of modeling, architecture, and Quality-of-Service issues in high-speed networks. In particular, he is interested in developing techniques to map network level to application level performance measures.

Dr. Guérin is a member of Sigma Xi and the IEEE Communications Society, and is an editor for the IEEE Communications Magazine and was an editor for the IEEE Transactions on Communications. His email address is guerin@watson.ibm.com.

Asad Khamisy (S'91) received the B. Sc. degree in computer engineering, the M. Sc. and the D. Sc. degrees in electrical engineering from the Technion–Israel Institute of Technology in 1988, 1991 and 1994, respectively.

Currently, he is on the faculty with the Electrical Engineering Department at the Technion. His research interests include high-speed local and wide area networks, and performance analysis. His email address is asad@tx.technion.ac.il.