A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic

Vincent Rosolen*, Olivier Bonaventure† and Guy Leduc

Research Unit in Networking, Montefiore Institute, University of Lège, Belgium.
E-mail : leduc@montefiore.ulg.ac.be
Alcatel Alsthom Corporate Research Center, Antwerp, Belgium
Now with the University of Namur (FUNDP), Belgium.
E-mail : Olivier.Bonaventure@info.fundp.ac.be

Abstract

In ATM UBR networks supporting TCP traffic, optimal efficiency can only be envisaged if switches adopt a discard mechanism that operates at the packet level rather than the cell level. In this paper, we define a variant of the RED discard strategy suitable for ATM switches. An interesting feature of this ATM-RED is that it has a similar per-VC implementation complexity as the Early Packet Discard (EPD) algorithm. To study the efficiency of the ATM-RED discard strategy, we compare its performance with plain the UBR, EPD and Fair Buffer Acceptance (FBA) discard strategies by means of simulation with TCP/IP traffic. We give comparative results with respect to different performance criteria such as goodput and fairness in various environments, such as end-to-end ATM networks and IP-based networks with an ATM backbone, in both single-bottlenecked and GFC topologies.

1 Introduction

In the Internet, non real-time applications that require reliable delivery use TCP, which is a complex protocol meant to control the flow of IP packets in the network and recover from packet losses and duplications. These applications are characterized by a highly variable traffic, little or no sensitivity to end-to-end delays and delay variations, and no need for guaranteed bandwidth.

ATM networks have been designed to support efficiently a large range of services at a reasonable cost, from real-time interactive applications to non real-time bulk data transfer. In the latter category the UBR ATM Transfer Capability (ATC) is a best-effort service whose quality is comparable to IP, except that ATM is connection-oriented and preserves the ordering of ATM cells. Therefore, running TCP over ATM UBR is likely to behave roughly like TCP over IP and be suitable for non real-time applications. Combined with the simplicity of UBR compared to other ATCs, this makes UBR a very attractive solution to support TCP traffic, and is clearly the reference point to which other more sophisticated ATCs should be compared.

It is well-known that the behaviour of TCP over IP or TCP/IP over UBR is improved by adding discarding mechanisms in routers or switches [RF95]. However, in this context there is an important difference between IP and ATM, viz. the Protocol Data Units (PDU) dealt with in these devices. Discarding mechanisms in ATM work basically at the ATM cell level, whereas they work at the packet level in IP. However, as discarding an isolated ATM cell is as costly as discarding a series of cells from the same IP packet, it is clear that ATM cell-discard strategies that would ignore the higher level PDU boundaries would lead to poor performance. Therefore, at the price of some layering violations, several such cell discard strategies have been proposed and studied. In this paper we extend these results.

We define a variant of the Random Early Detection (RED) [FJ93] discard strategy, initially proposed for IP routers, which is suitable for ATM switches. This algorithm has been implemented in the STCP simulator [Man96] and its performance assessed and compared to several other algorithms, namely Early Packet Discard (EPD) [RF95], Fair Buffer Allocation (FBA) [HK98] and Tail Drop, *i.e.* standard UBR. As performance criteria, we study the TCP goodput and the fairness between TCP connections, as well as the link utilization in several environments, including end-to-end ATM networks, router-based architectures, asymmetrical access and GEO satellite links.

2 Cell discard strategies

Several techniques have been proposed to improve performance in ATM switches. Namely, dropping policies such as Tail Drop, Drop From Front [LNO96] and variants with relation to the data unit format (*i.e.* cell or frame) were

developed early to help switches deal with congestion. All these methods were partially successful, in the sense that they did achieve acceptable performance on congested links, but lacked fairness when throughputs were analyzed. Further investigation on this issue led researchers towards schemes aiming at both levels of performance.

2.1 Early Packet Discard (EPD)

EPD was first proposed in [RF95] as an improvement over Partial Packet Discard (PPD) and the default tail drop cell discard policy. The idea behind EPD is that a discarded cell makes its corresponding AAL5-PDU incomplete, and therefore useless : cells from this packet continue to flow even though the entire AAL5-PDU will have to be retransmitted. This major problem is the *orphan cell syndrome* and can lead to severe throughput degradation. To improve this situation, when the buffer occupancy of an EPD switch reaches a fixed threshold ( ), the switch drops entire AAL5-PDUs instead of dropping individual cells. The threshold can be seen as a parameter through which overflow is more or less conservatively prevented, but also allows a few entire packets to pass their way undamaged even when congestion is experienced.

EPD has already been extensively discussed in the literature [Tur96, CT97, KKTO97, RF95], as well as numerous variants and improvements. As a result, this method has been widely implemented in commercial ATM switches.

From an implementation point of view, EPD is more complex than the default tail drop cell discard policy since with EPD the switch needs to maintain two bits of state for each individual VC to support EPD. The tail drop discard policy does not require any per-VC state and the PPD discard policy requires one bit of per-VC state.

2.2 Fair Buffer Allocation (FBA)

While EPD greatly improves throughput results, it does not attempt to improve fairness between the competing VCs. The reason is that the EPD scheme keeps track of *per-VC states* to implement the policy. With *per-VC accounting* , one could also keep track of each VC by counting the number of cells from each VC in the buffer. This observation is the basis of Selective Packet Dropping (SPD) and allows better (fairer) allocation of the buffer resources to the active VCs.

In this scheme, an AAL5-PDU from one connection is discarded if the buffer occupancy reaches a given threshold , *and* if this connection takes more than its fair share of the buffer. This simple algorithm is not yet quite satisfactory, since buffer occupancy can remain low. Indeed, a particular connection that has reached its share of the buffer will remain stuck to this allocation, even if the other active connections use little of their respective share. A better result can be achieved with the FBA algorithm [HK98], which proposes a smoother but slightly more complex scheme. Instead of rejecting the first cell of an AAL5-PDU as soon as the threshold is reached and the fair share is exceeded, the switch allows greedy connections to exceed their fair share, with respect to a rejection function such as illustrated in figure 1. In this figure, we show the *normalized* maximum share (with respect to the fair share) of a VC versus the buffer occupancy. Thus, a share of 1 indicates that the VC has cells in the buffer, where is the total buffer size (in cells) and is the number of active VCs[1].

Whenever a first cell of an AAL5-PDU for a particular VC arrives, the corresponding share of this VC is evaluated and compared to the limit expressed by the normalized maximum share for the current buffer occupancy (figure 1). If the calculated value exceeds the limit, the entire AAL5-PDU is discarded. The FBA algorithm also introduces a *scaling factor* ( ) to increase the flexibility of parameter tuning. The effect of this parameter, which must be between 0 and 1, is to roughly shift the curve down as the scaling factor decreases to zero, enabling the algorithm to behave more or less conservatively with relation to the total buffer size.

From an implementation point of view, FBA forces the switch to maintain, for each established VC, a counter to count the number of cells belonging to each VC that are currently in the switch buffers. The size of this counter is obviously a function of the size of the switch buffers. For example, a switch with a 16k cells output buffer would need to maintain a 14 bits counter for each established VC. With 32k established VCs or even more, this may lead to a large amount of memory only to implement a discard strategy.

2.3 Random Early Detection (RED)

The RED scheme was first proposed in [FJ93]. Its objective is to provide a control the average buffer occupancy in order to provide a fair bandwidth allocation, along with a simple implementation. The algorithm relies on an approximation of the *average* queue size in order to improve the buffer utilization through *low average buffer occupancy*, and can be summarized as follows :

- The average queue size is estimated through an exponential weighted average with weight :

  (1)

  where is the *instantaneous* queue size. refers to a time granularity which is mandatory for this sort of calculation. This formula can be seen as a low-pass filter through which the signal "instantaneous queue size" passes, yielding the output "average queue size". is the time constant of the filter.

  If remains under a fixed threshold , no discarding occurs.

  If exceeds , discarding must occur on each arriving data unit with a probability . This probability

[1] A VC is said to be *active* in an ATM switch if it has at least one cell buffered in the switch.

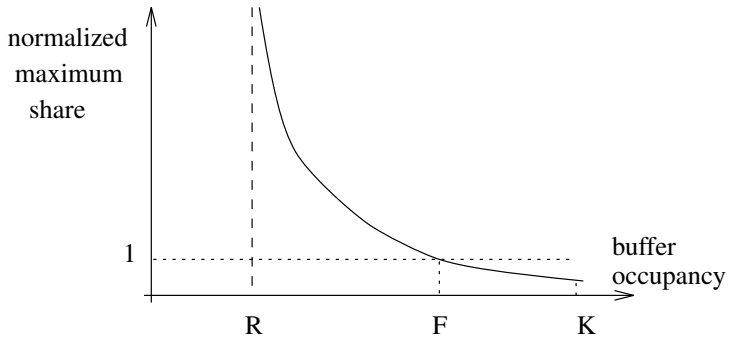

Figure 1: Rejection function in the FBA algorithm.

increases with and is a function of , a second fixed threshold , and , which defines the "slope" of the first part of the dropping probability function. The specific function chosen for this paper is given in figure 2 and was proposed in [Flo97].

In addition to maintaining a low average buffer occupancy, the RED algorithm gives an elegant answer to the *global synchronization* syndrome: the probabilistic approach allows routers to discard packets roughly in proportion to the connection's share of the bandwidth through the router. This ensures that if multiple discards must be made, they will probably concern the greediest connections; thus, it is unlikely that all connections see one of their data units discarded, which avoids simultaneous beginnings of slow-start phases. The active queue management technique provided by RED has proven to be efficient and as a consequence, RED has been recommended as the default queue management mechanism in legacy routers [BCC+98].

2.3.1 ATM-oriented implementation

RED's original version concerns IP routers and thus deals with *packets*. While seeming an attractive strategy, RED needs to be adapted to be used in ATM switches. In this section, we propose a specific implementation of RED for ATM switches. Our implementation of RED (ATM-RED) is based on two principles. First, we want to either accept or discard entire AAL5-PDUs. Thus, we need a way to identify the boundaries of the AAL5-PDUs. Second, we want to be as close as possible to the original RED algorithm, but with a minimum complexity.

The algorithm is implemented as a four states finite state machine (FSM). When a cell is received in the buffer, this FSM decides on the basis of its current state, the position of the cell in the AAL5-PDU (first or middle cell : AUU=0 or last cell : AUU=1) bit and the dropping probability whether this cell should be accepted ( ) in the buffer or discarded ( ). The complete FSM is shown in figure 3. In this figure, indicates that the arriving cell is the last cell of an AAL5-PDU (AUU=1) and indicates that the received cell is a normal cell (AUU=0). The algorithm computes the dropping probability (in RED) in two states (*accept-first* and *accept-cell*), when the average buffer occupancy is above and every time a cell is received. We use when the dropping probability indicates that the subsequent AAL5-PDU should be dropped and otherwise. If the computed probability indicates that the cell should be dropped, the algorithm will still accept the end of the AAL5-PDU but will drop the subsequent AAL5-PDU of this VC.

The initial state in which the switch lies is *accept-first*, as represented on the figure with a dotted arrow. The algorithm is always in the *accept-first* state upon the arrival of the first cell of an AAL5-PDU. When there is no congestion ( ), the FSM stays in either the *accept-first* or *accept-cell* state and all the received cells are accepted in the buffer. If congestion occurs ( ), the algorithm will compute a dropping probability and will randomly discard an entire AAL5-PDU. The discarding of an entire packet can be visualized by the transition in state *discard-cell*: the state cannot change until an "AUU=1" cell arrives.

The main feature of this adaptation is that if a *first* packet discard decision is made, it concerns the *next* packet (with relation to the cell that caused the decision). Indeed, it is unlikely that whenever the discard probability leads to a discard, this cell is the first one of its corresponding AAL5-PDU. Thus, the discarding of an *entire* packet can only be achieved on the next packet, because the decision is always taken on an arriving cell basis. The state machine must then pass through *discard-next* before getting to *discard-cell* where cell discard eventually occurs. Note that if the first marked cell happens to be the first cell of a corresponding packet, discarding can take place immediately, as shown by the transition from *accept-first* to *discard-cell*. Finally, if congestion is still present in the network after the first packet discard, the algorithm will continue to decide that packets should be discarded, and it may be possible that a series of packets be discarded: the state machine will hop between states *accept-first* and *discard-cell*.

In the original RED, the probability to drop a packet is computed at the arrival of each packet as a function of the

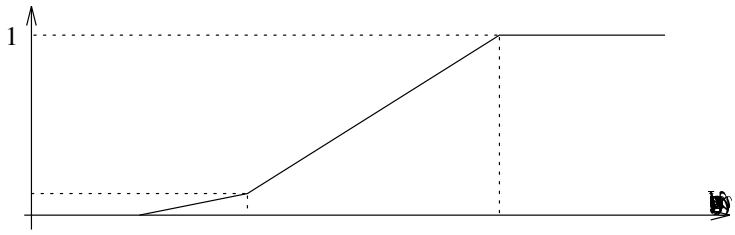

Figure 2: Dropping probability function in the RED algorithm.

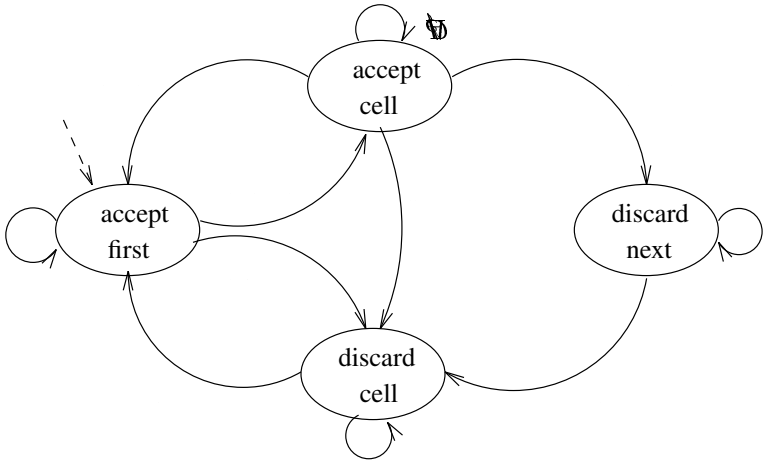

Figure 3: ATM-RED state diagram.

packet size, the maximum dropping probability ( ) and the average queue size. In an ATM switch, we cannot determine the size of the arriving AAL5-PDU when we receive the first cell of this AAL5-PDU. Thus, we cannot compute the dropping probability as a function of the packet size. However, we would still like the dropping probability to depend on the size of the AAL5-PDUs so that VCs with large AAL5-PDUs have a higher dropping probability than VCs with smaller AAL5-PDUs.

Actually, calculating this *cell* probability on each arriving cell allows us to obtain a larger value for the overall *packet* probability when packets are larger, and this is done without evaluating the packet size. This comes from the fact that, if $_{C}$ is the cell dropping probability and is the number of cells inside one packet, the corresponding packet dropping probability is given by

(2)

and is approximately equal to if $\ll$. Moreover, it can be shown that a dropping probability function (at the cell level) such as the one illustrated in figure 2, which is the one we implement, yields another function at the packet level. An example is shown in figure 4, which illustrates the three probabilities as a function of the buffer size : at the cell level and at the packet level (for two different MSS). The parameters that were used are $_{0}$, and $_{2}$ where is the maximum cell dropping probability.

This new *packet* dropping probability function is not only dependent on the MSS, but also has interesting properties with regard to the general behaviour of the algorithm. Among these features, the new function has a shape which is close to the one recommended in [Flo97] but considered too complex to be implemented. Here, this complex shape is automatically obtained at the packet level by adopting the simple shape at the cell level.

The main feature of our proposed ATM-RED algorithm is that it has the same per-VC complexity as the EPD algorithm since the switch only needs to maintain two bits of state for each established VC. However, from an implementation point of view, the price for this low per-VC complexity is the requirement to compute, on the worst case, a probability upon the arrival of every cell. A switch which only implements EPD does not obviously need to compute such

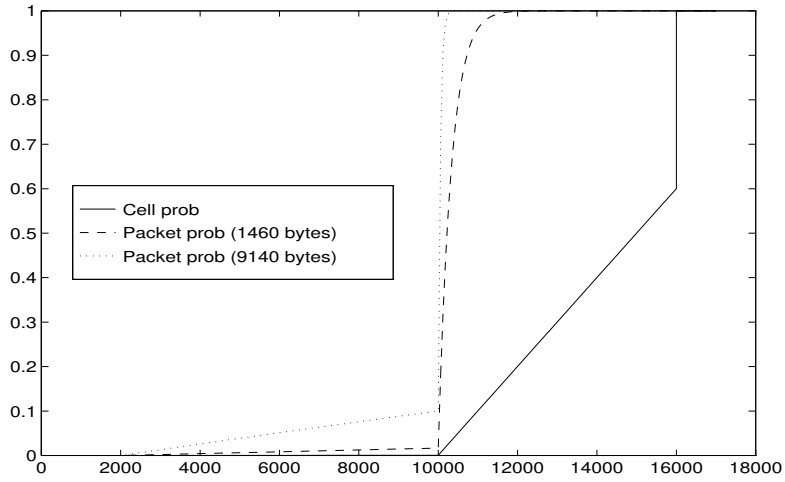

Figure 4: Discard probabilities at the cell and packet level as functions of the buffer occupancy (in cells)

probabilities.

Two implementations of RED for ATM switches have been proposed previously in [EA97]. The first implementation, named "cell-based RED" in [EA97] is equivalent to the classical packet-based RED algorithm except that a drop probability is associated with each arriving cell and once a cell has been dropped, the remaining cells of the AAL5-PDU, except the last one, are dropped as well as in the PPD algorithm. Another modification is that the average buffer occupancy is not estimated upon each cell arrival, but only upon the arrival of the first cell of an AAL5-PDU. From an implementation point of view, the complexity of this "cell-based RED" is equivalent to our ATM-RED, but the "cell-based RED" discards tails of AAL5-PDUs when congestion occurs while our ATM-RED always discards entire AAL5-PDUs. The second implementation proposed in [EA97] under the name "P-RED" is in fact a combination between the classical RED algorithm and FBA. Like the classical RED algorithm, P-RED measures the average buffer occupancy and probabilistically discard arriving AAL5-PDUs when congestion is detected. However, instead of computing the packet discard probability based on the size of arriving AAL5-PDU (an information that is only available when the last cell of the AAL5-PDU has been received), P-RED computes the packet discard probability based on the fraction of the buffer which is used by this VC. With this feature, P-RED is close to the F-RED algorithm proposed in [LM97] for IP routers. From an implementation point of view, P-RED is more complex than our ATM-RED algorithm since it forces the switch to maintain a counter for each established VC like FBA.

Finally, our ATM-RED algorithm could be used to notify congestion rather than to drop packets. This would require to mark ATM cells from entire AAL5-PDUs with the EFCI bits and propagate this notification to the IP layer at interface routers as suggested in section 8 of [RF99].

3 Simulations

All our simulations were carried out with the *STCP simulator* (developed by Sam Manthorpe at the EPFL [Man96]), which includes the complete BSD 4.4 TCP/IP implementation. This version of TCP includes the slow-start, congestion avoidance, fast retransmit and fast recovery as well as the RFC1323 timestamp and large windows extensions. We have patched it with the SACK implementation available in [Mah96] to support the new selective acknowledgements option. The source code of STCP has been modified to include the FBA and ATM-RED discard methods described in the previous section. For the sake of easier interpretations, the following assumptions are made in all types of simulated environments (or "*scenarios*") :

- All sources are assumed to be identical with respect to their *equipment*. In other words, features such as interface cards, link delays and bandwidth, are unique for one type of environment, unless specified.

  The sources are fileservers based on an on-off model, with a null off-period. This type of source is more realistic than the infinite source model, with respect to common applications using TCP/IP. For instance, elements like TCP's slow-start algorithm [Jac88] have a non-negligible impact on simulation results. All our simulations were run for an amount of time designed to have the sources successfully transmit a dozen files to reach steady state in network statistics.

  The queues that model the switches' buffers have a unique size, which is fixed at 16000 cells. The reason for this choice is that easier comparisons can be made between the three discard methods, regardless of buffer resources. The choice of 16000 reflects fairly well what is implemented in most of today's ATM switches.

- TCP's timer granularities values have been chosen in order to fit with modern TCP implementations, *i.e.* 200 ms for the slow time-out granularity and 50 ms for the fast time-out granularity.

3.1 Environments

ATM networks are intended to be widely developed, and are supposed to support a large number of applications in all types of environments. As a consequence, the more flexible a particular switch, the better behaviour it will exhibit. This holds particularly for the TCP/IP protocol suite, whose share of global networking keeps growing [TMW97]. This flexibility is thus an important feature of the methods evaluated in this paper, and this is the reason why we consider numerous different environments. The main differences between the corresponding scenarios are based on three characteristics of the simulated environments : architecture, topology, and access method. The two possible architectures are ATM end-to-end and backbone ATM ; the two possible topologies are single bottleneck and generic fairness configuration [Sim94] ; and the two possible access methods are direct access and asymmetrical subscriber line. These environments are described in the next sections.

3.1.1 Direct access on a single-bottlenecked ATM network

Our first simulation model consists of a single bottleneck link between two switches, as shown in figure 5.

This simple topology can be used as a basis for evaluation. Besides, this model has often been used in numerous previous analyses [GJKF98], [EA97]. The characteristics of this environment are the following :

the bottleneck is shared by 20 pairs of source/destination workstations in a bidirectional fashion (10 sources on each side of the network) ;

all links are OC-3 type links (155 Mbps) ;

the main bottleneck is characterized by a 10 ms delay, while the workstations are organized in two *clusters* of five source/destination pairs on each side, each cluster being characterized by either 2.5 or 10 ms access links ;

the TCP sources transmit files of 5 MBytes and use a 2 MBytes window, which is larger than the bandwidth-delay product for this environment.

3.1.2 Asymmetrical access on a single-bottlenecked ATM network

The asymmetrical access-related model is basically identical to the one described above, as shown in figure 6.

The network is again modelled by a single bottleneck between two switches. The main differences concern delays and bandwidths. We have chosen to simulate an asymmetrical environment, close to the one encountered in xDSL access networks. ATM is indeed currently deployed as a backbone technology for such wide area networks. We consider here characteristics that are typical of ADSL environments :

the sources are assumed to be servers attached to legacy 10 Mbps Ethernets (left side of the figure) ;

the link has a bandwidth of 34 Mbps with a delay of 10 ms ;

the destinations are assumed to be ADSL clients with access links of 200 kbps upstream (from end system to network) and 2 Mbps downstream, and the ADSL modems introduce a delay of 10 ms (right side of the figure) ;

the TCP sources transmit files of 512 kbytes, and the TCP window is set to the standard 64 kbytes, which is enough in this environment.

Note that as opposed to the previous scenario, this environment is unidirectional in nature.

3.1.3 Direct access on a GFC-shaped ATM network

The next environment we consider involves several switches and bottlenecks, as well as different groups of sources (see figure 7). This model inspired by the Generic Fairness Configurations that were used by the ATM Forum while developing the Available Bit Rate (ABR) service category. It is a sufficiently good approximation of a meshed network to allow more general conclusions. The following assumptions are made :

the three bottlenecks have a bandwidth of 31, 155 and 93 Mbps respectively, and are characterized by a delay of 10 ms ;

the transmitted files have a size of 1 MByte ;

the maximum TCP window is opened up to 512 KBytes, in order to allow the sources to fully utilize the available bandwidth ;

all sources are organized in *clusters* of 10.

the sources experience a 2.5 ms delay before reaching their switch ; the corresponding links have a bandwidth of 155 Mbps.

This configuration has been specifically designed to obtain the following scheme :

the main bottleneck is the middle link (*i.e.* between switches 2 and 3), which is shared by clusters , and . The other two clusters (and) have a perturbative function (cross-traffic).

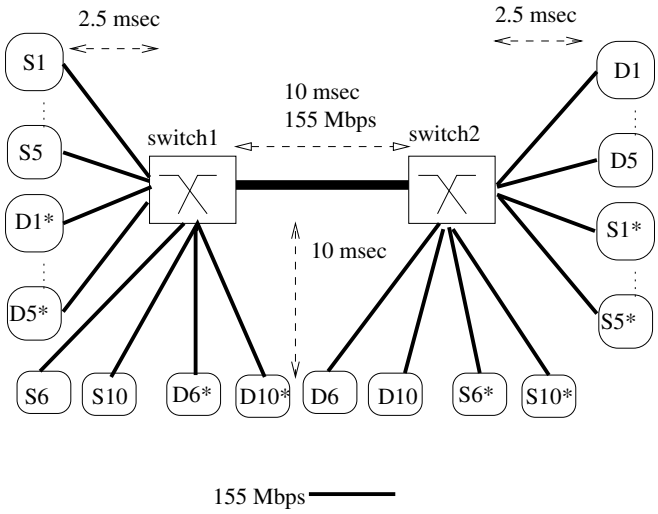

Figure 5: Direct access on a single-bottlenecked ATM network

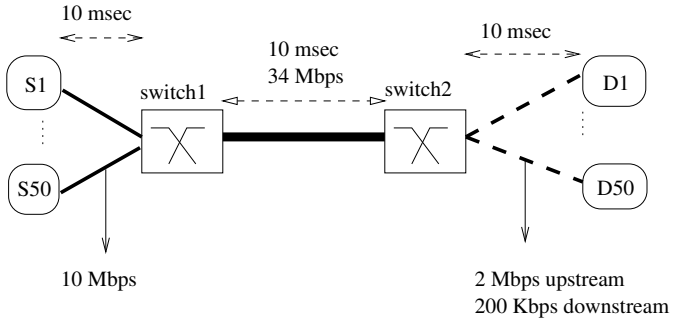

Figure 6: "Asymmetrical access on a single-bottlenecked ATM network" simulation model.

- the number of each type of source (clusters) and the link bandwidths are fixed such that the cluster should get 30 % of the main bottleneck, the cluster 10 % and the cluster 60 %. If we choose a 155 Mbps bottleneck link, and if the number of sources is known, then this scheme gives the bandwidths of the left and right links (respectively 31 Mbps and 93 Mbps) as follows. The and clusters act as perturbative traffic mainly for clusters and respectively, and should normally get the same respective bandwidths, namely an equivalent of 10 % and 30 % of the main bottleneck link.

The same principle will be applied in the following sections when a similar topology is considered, but with a different architecture and/or access method.

3.1.4 Asymmetrical access on a GFC-shaped ATM network

As in section 3.1.2, we consider a variant of the previous scenario, with respect to the access method used by the workstations. The model of this scenario is basically the same as the one depicted in figure 7 ; the only feature that we modify concerns the various bandwidths and delays :

all delays have a value of 10 ms ;

the access bandwidths are respectively 2 Mbps (symmetrical) at the source and 200 kbps up/2 Mbps down at the destination ;

the ATM links are characterized by bandwidths of 10, 50 and 30 Mbps from left to right in figure 7.

In addition, the number of workstations per cluster is increased to 25 for clusters , , and , and to 50 for cluster , since this environment attempts to model a "crowded" network, as we can expect with ADSL technology.

3.1.5 Router access on a single-bottlenecked ATM backbone

In the following sections, we will refer to an *ATM backbone* as an IP-based environment, in which routers have a signifi-

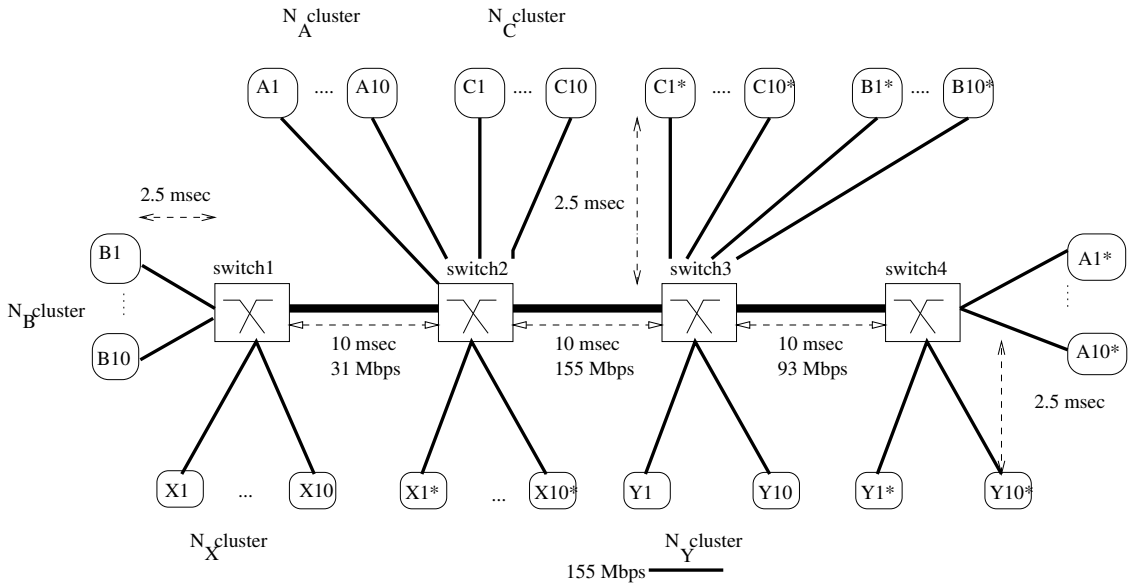

Figure 7: "Direct access on a GFC-shaped ATM network" simulation model.

cant role. The ATM switches will be hidden from the workstations by routers that act as access points to the ATM backbone. Such an architecture is typical of what can be seen in today's Internet Service Providers (ISP) structures – several local points of presence (POP) and Network Access Points (NAP) organized in a hierarchical way.

To represent a router access on a single-bottlenecked ATM backbone, we replace a workstation in figure 5 by a router to which a cluster of Ethernet workstations is attached. This environment is organized as follows :

- the ATM backbone link (between the two switches) has a bandwidth of 34 Mbps ;

  two clusters of five routers each are attached to each ATM switch, each cluster being characterized by an access delay to the switch of 2.5 ms and 10 ms respectively ;

  the link between a router and a switch has a bandwidth of 34 Mbps ;

  a cluster of ten workstations is attached to each router, by means of an individual 10 Mbps link with a 1 ms delay ;

  there is a total of 50 sources on each "side" of the network.

In this scenario, TCP sources transmit files of 500 KBytes, and the maximum TCP window is set to 128 KBytes. Note that the communication is bidirectional as in the scenario described in section 3.1.1.

Routers

To support this environment with STCP, we had to sligthly modify it in order to support routers since the STCP simulator does not directly provide any router component. In our work, routers were implemented as *VC merging points*. Such a merging performs a partial reassembly of the received cells so that the cells corresponding to a single AAL5-PDU are transmitted back-to-back in sequence. A typical router will thus be modeled by two queues as shown in figure 9.

The input links are attached to the first queue, which behaves as the merging point. This queue is attached to the output buffer (the second queue), where the queuing of AAL5-PDUs actually occurs. The speed of the link between the merging point and the output buffer is assumed to be infinite by the simulator : as soon as a complete AAL5-PDU has been received at the merging point, it is moved in zero time to the output buffer.

The reasons why we chose to develop VC merging points – instead of a complete router model – are based on the observation that the only influence of a router, at least in the simulation environments that we consider in this paper, is to aggregate the traffic from several TCP sources on a single VC. In our simulation environments, queueing does not occur inside the routers and no packets are discarded by the routers. Thus, the partial reassembly and the back-to-back transmission are sufficient to emulate the role played by a router, whose complete implementation would have otherwise proved exceedingly code-consuming.

All the following scenarios are derived from the four depicted in the previous four sections. The main architectural difference is that instead of being directly attached to ATM switches, workstations are connected to routers, which in turn have direct access to the ATM backbone. Worksta-

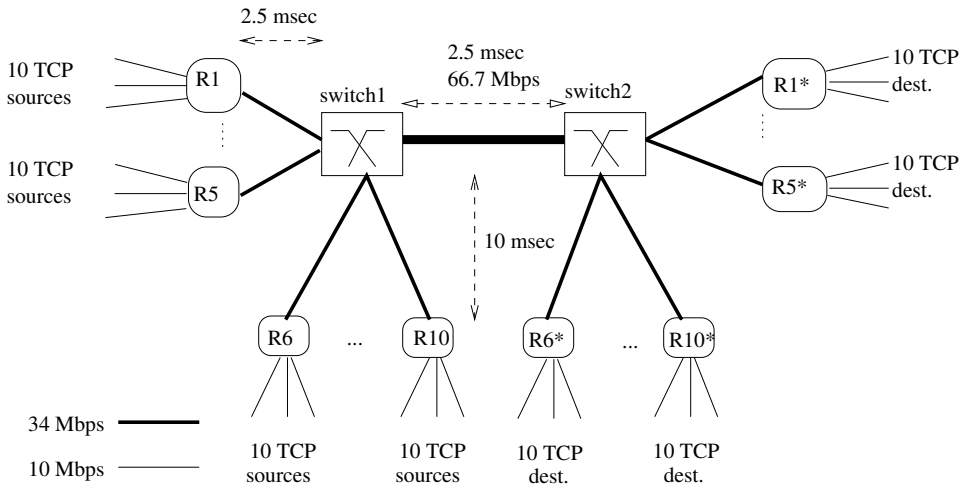

Figure 8: "First router-based scenario" simulation model.

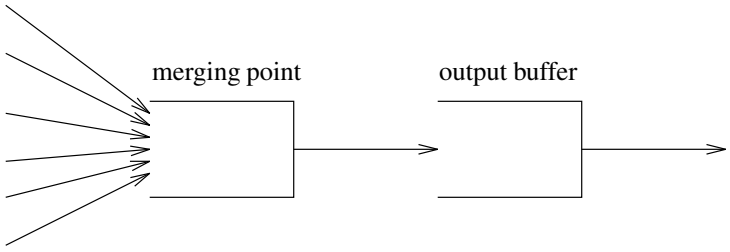

Figure 9: Router model.

tions will be organized again in clusters attached to a single router, and clusters of routers will be attached to a single ATM switch.

3.1.6 Asymmetrical router access on a single-bottlenecked ATM backbone

This environment has various elements from scenarios described in sections 3.1.2 and 3.1.5. We keep the topology and the hierarchical organization of scenario 3.1.5, while we introduce the unidirectional nature and the asymmetrical bandwidths of scenario 3.1.2 :

- 5 routers are attached to each switch ;

  25 workstations are attached to each router ;

  the TCP sources transmit files of 500 kbytes ;

  the delays are 2 ms for the workstation-router links, and 10 ms for all other links ;

  the bandwidths are 34 Mbps for the router-switch links and the backbone link, 10 Mbps at the source and 200 kbps (up) / 2 Mbps (down) at the destination.

Note the slight difference with scenario 3.1.2, in which the sources' bandwidth was 2 Mbps. Indeed, in a pure ATM network, we assume that the 2 Mbps bottleneck is known by the source when the ATM contract is negotiated between source and destination. This assumption is no longer relevant in IP-based architectures, since the ATM contract does not "reach" the end stations.

3.1.7 Router access on a GFC-shaped ATM backbone

For this environment shown in figure 10, we replace again each workstation of scenario 3.1.3 by a cluster of workstations attached to a router. Namely, clusters [illegible] through [illegible] in scenario 3.1.3 are here clusters of routers, or "superclusters" (groups of routers to which are attached groups of workstations). This scenario is organized as follows :

the three bottlenecks have bandwidths of 9, 45 and 27 Mbps respectively, and are characterized by a delay of 10 ms ;

there is a total of 5 routers per cluster of routers ;

the delay between a router and a switch is 2 ms ;

all sources are attached to a router by clusters of 10 ;

the transmitted files have a size of 500 kbytes ;

- the sources experience a 1 ms delay before reaching their corresponding router ; the corresponding links have a bandwidth of 10 Mbps.

3.1.8 Asymmetrical router access on a GFC-shaped ATM backbone

This scenario is basically identical to the previous one, the only differences concerning the various bandwidths :

> the three bottlenecks have bandwidths of 10, 50 and 30 Mbps respectively ;
>
> the router-switch links still have 34 Mbps, while bandwidths are now 10 Mbps at the source and 200 kbps (up)/2 Mbps (down) at the destination.

3.1.9 Satellite environments

In addition to the eight environments previously described, we study the effect of GEO-type satellite links in each corresponding "direct access" scenario. To model this additional feature, we simply change one of the link's delay, to obtain a GEO link in the particular scenario. Namely, a 250 ms delay (up and back) is given to

> the "main" ATM link in non-GFC scenarios – that is, the one which provides a path between the two ATM switches in scenarios described in sections 3.1.1, 3.1.2, 3.1.5 and 3.1.6 ;
>
> the "middle" bottleneck in GFC scenarios – that is, the link between the second and third ATM switches in scenarios described in sections 3.1.3, 3.1.4, 3.1.7 and 3.1.8. We do not consider more GEO-type links in these environments, since the delay introduced by a single satellite is big enough to dramatically increase the round-trip time and disturb the TCP connections.

Again, for these GEO environments, the TCP window used by the sources is adapted in order to avoid any bottleneck at the source level.

Taking into account these additional four scenarios, we have a grand total of twelve distinct environments, whose simulations are analyzed in sections 4.

3.2 Parameters

In the following sections, we will refer to simulations that were successively run with each of the four packet discard options described in section 2. With the exception of plain UBR, these strategies introduce parameters that need to be fixed. In EPD, the threshold $\perp$ we consider is set to 14000 cells (roughly 90 % of the buffer size) in order to efficiently use the buffer resources. In the same fashion, FBA's threshold $\mathcal{B}$ and scaling factor Σ are set to 14000 cells and 0.9 respectively, since this choice seems to stand out as mentioned in [GJKF98] and [RBL98]. As for RED, we must remain careful when choosing *cell* related parameters which lead to different quantitative *packet* related results, as described in section 2.3. The values that were chosen for parameters $\mathcal{B}$, $\mathcal{B}$, $\mathcal{B}$ and are 0.001, 0.1, 2000 and 12000, respectively.

Finally, for the MSS size, we consider the two values of 1460 and 9140 bytes in our simulations, since 512 bytes becomes less frequent in most actual networks. Moreover, the size of 9140 bytes is used only in end-to-end ATM environments (*i.e.* "ATM networks" scenarios in section 3.1), whereas 1460 bytes is used only in ATM backbones, since a value of 9140 bytes is no longer relevant in IP-based architectures.

3.3 Evaluation criteria

The overall performance of a given discard method has several aspects. The most obvious one is to improve resource utilization in avoiding the retransmission of useless cells. Nevertheless, one must also take into account undesired effects which could result from the chosen scheme. Indeed, a certain discard method maintaining high throughput under poor utilization conditions would, for example, be useless. This type of situation can occur in a network whose switches do not implement any discard strategy : in this case, we would expect useless cells to pollute bandwidth, which would result in possible excellent throughput but very low efficiency. The choice to investigate multiple evaluation criteria also completes the choice to simulate different environments : the best method is the one that behaves correctly, in the sense of the largest number of criteria, and under the broadest range of situations. These are the reasons why we address here three performance issues : efficiency, throughput and fairness. The end-to-end delay was not considered essential, because TCP applications are not interactive. Anyway, the average queueing delay is easily computed from the average queue sizes, and the transmission delay can be calculated, so that the end-to-end delay can be derived easily.

3.3.1 Throughput and efficiency

The most important end-to-end parameter remains the throughput achieved by a connection. In all our simulations, we analyze the *goodput* of each TCP source ; that is, the TCP bytes successfully acknowledged in the simulation time. This choice allows us to give interpretations at a TCP level, which guarantees a minimal relevance with pure applicative throughputs. Moreover, *ideal values* can be defined in taking into account protocol overheads, and give a precise idea of a given strategy's overall behaviour. Considering the goodput reflects thus also the first mentioned criterion (utilization or efficiency) : this presents the asset to show two features inside one single result. Note that considering TCP data for throughput *and* efficiency calculations reflects not only "how the pipe is filled with TCP data" but also "how *well* the pipe is filled with *useful* TCP data".

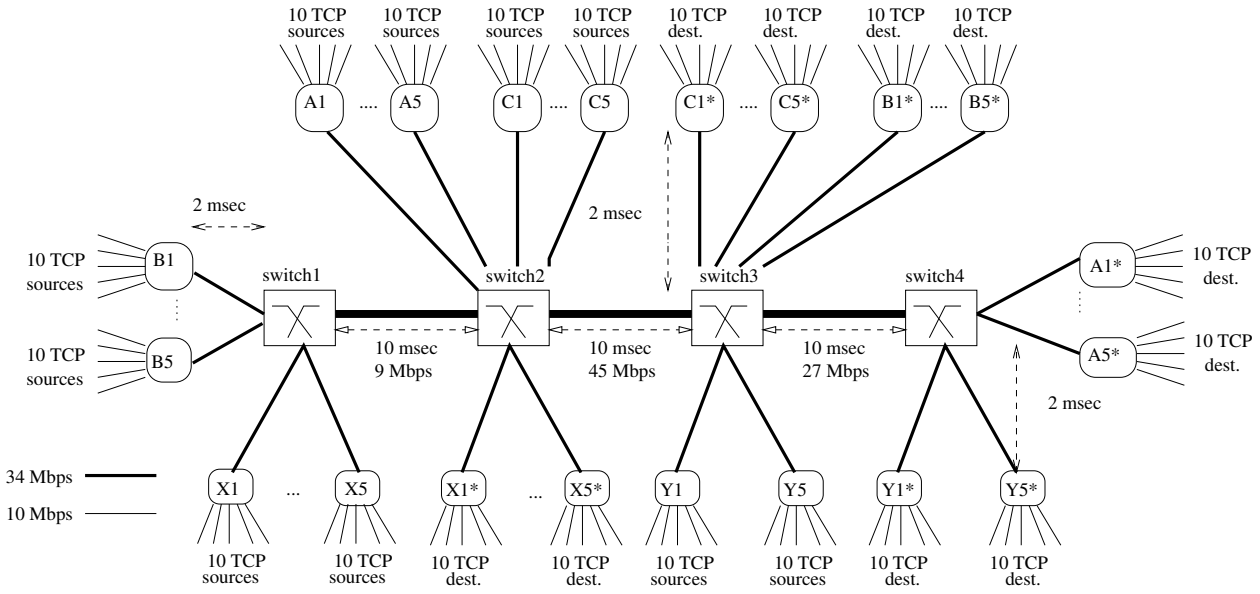

Figure 10: "Direct acces on a GFC-shaped ATM backbone" simulation model.

We present goodput results in a *normalized* fashion ; that is, any actual goodput result has been divided by the corresponding *ideal* goodput. To compute the ideal goodput for sources, we consider that the bottleneck link is used at 100 % and the bandwidth is fairly shared by all the sources. The actual goodput can thus exceed the ideal goodput. Moreover, the latter is obviously environment dependent, and that is the reason why the average ideal goodputs are described at the beginning of the sections related to simulations. Each result thus reflects an *average* taken on all the local goodputs, *i.e.* we have the definition for the normalized average goodput :

$$\text{normalized average goodput} = \frac{\text{average effective goodput}}{\text{average ideal goodput}} \quad (3)$$

which allows us to always have a target result of 100.

For IP-based environments where many sources are connected to a router, the goodput we consider is the goodput of the aggregated TCP sources, that is the goodput of all the sources that share a common ATM VC.

The fact that the goodput has been so normalized, has no impact on the comparison between the 4 algorithms: comparing absolute values or normalized values (whatever the normalization is) leads to the same conclusion. However, our normalization brings an additional information: namely, it states how far the results are from the ideal case, and no just which proportion of the bottleneck link is used. This is particularly suitable when different clusters have different bottleneck links, like in GFC scenarios.

3.3.2 Fairness

The macroscopic behaviour of the simulated environment can be visualized by means of a fairness index, which we define as follows :

$$\quad (4)$$

where is the number of sources and is the *fairness coefficient* of source in terms of the average throughput , *i.e.*

$$\quad (5)$$

Concerning the definition of , the idea is to have a global result which conveys all the local results. If we express the latter by means of a deviation from a mean (and ideal) value, we implicitly penalize more heavily the throughput distributions that exhibit a more pronounced scattering. Thus, when all sources get the same fraction of the bandwidth, for all , and the ideal fairness index is obtained with . If for example, we have two sources that get goodputs of 10 Mbps and 20 Mbps respectively, the corresponding fairness index would be ; with goodputs of 5 and 25 Mbps, this index falls down to 33.33. Note that for these two examples, the mean goodput is the same, but the second situation yields a worse index due to the more pronounced scattering. Moreover, if for all , then .

The definitions chosen here aim merely at expressing *relative* results and does not yield results which depend upon ideal *absolute, environment-dependent* values for the goodput. The latter are best expressed through efficiency results

such as those defined in the previous section. More precisely, as defined in (4) and (5), a fairness index is independent of the *expected* goodput : it only depends upon the *mean* goodput. We think that this choice is more pertinent, since the goodput and fairness related results follow orthogonal ways of interpretation.

Finally, we could think that there is a redundancy between these fairness indices and the fact that the ideal goodput results, because the latter have been normalized by taking account of their ideal fair share. Howeve, this is not so, because the fairness indices refer to the fairness (i.e. the fairness among sources), whereas the ideal goodput takes account of the fairness (i.e. the fairness among clusters).

4 Results

In addition to the EPD, RED and FBA packet discard methods, we also consider standard UBR (with tail-drop policy) to better understand the benefits brought by a given discard strategy.

4.1 Overview

In the following sections, general results are presented with regard to each simulation scenario and discard method. As described in section 3.3, statistics such as goodputs, bandwidth utilization and fairness indices are collected and presented. Figure 11 presents the relevance of each value that will appear in the following sections.

On an end-to-end point of view, *workstation statistics* are analyzed by means of the average goodput, as well as the fairness index. These two values are related to a *cluster* of workstation, *i.e.* a group of sources or destinations which have the same characteristics. For example, section 3.1.1 defines four different clusters of workstations : two on the left side and two on the right side of the network, with two different delays for two clusters on the same side. Figure 4 directly refers to this first scenario, but the same principle will be applied throughout all the other scenarios : workstation results will always be presented on a *cluster* basis.

Switch statistics are also presented for each known bottleneck in a particular scenario. We show the mean buffer occupancy, the maximum buffer occupancy, and the packet loss probability. Note that whenever a switch is "behind" a major bottleneck, its related statistics are not presented, since it does not lose any cells and thus cannot be an issue.

Finally, *link statistics* are presented for each ATM backbone link in the simulation model. We only show in this paper the bandwidth utilization that is achieved by all sources transmitting on the same ATM link. The bandwidth utilization is the ratio between the total number of transmitted application bytes (not counting retransmissions) and the maximum number of bytes the link could have possibly transmitted (taking protocol overhead into account).

As already mentioned, we express all our results in a normalized fashion. In other words, the values that will be presented below are expressed as a percentage of a certain ideal value, such that the target value is always 100. There are however three exceptions : the *packet loss probability*, the *mean buffer occupancy* and the *maximum buffer occupancy* must be as small as possible. Note that the latter two are expressed as a fraction of the buffer size, which is 16000 cells, to appear as a percentage as well.

To increase the readability inside the tables, we show the best results in a **bold** typeface, in order to have a quick peek at the best overall discard method. As a rule of thumb, the *global fairness index* and the *bandwidth utilization* are the most explicit features to concentrate on.

4.2 End-to-end ATM architecture

4.2.1 Direct access on a single-bottlenecked ATM network

Figures 12 and 13 and table 1 show a summary of the results collected for the scenario described in section 3.1.1. For this environment, the ideal goodput can be calculated ; taking into account protocol overheads, we obtain an ideal TCP goodput of 13.45 Mbps (10 sources sharing a 155 Mbps link).

The first major observation that can be made is that the overall results are rather homogeneous with respect to the discard method. Moreover, the results are close to optimum for the short-delayed cluster, while about half of what is expected for the long-delayed cluster. The overall performance is thus roughly between 70 and 80 %, as expressed by the bandwidth utilization index in table 1. Similarly, fairness tends to be slightly lower for the long-delayed cluster (figure 12).

Note that in figure 12, we can already see an example of a cluster of workstations "vampirizing" another. As explained in section 3.1.1, the value above 100 % for the mean goodput is not at all an anomaly ; it simply means that the mean goodput of the workstations belonging to this cluster is above what is *expected* (not *physically limited*). This phenomenon is typical of situations where different conditions apply to different clusters : for example, in figure 12, the larger round-trip time that is experienced by the second cluster suffices to create a strong inequity in resource allocation. As a result, the mean goodput obtained by the second cluster, in this case, is only 59 % of what is expected. Note that this inequity does *not* appear in the fairness indices, since each of these values is only relevant for one single cluster.

As regards the TCP goodput, a slight preference already appears for RED, which has the essential asset to keep buffer occupancy low, together with a low packet loss probability. Compared to other algorithms, RED thus reduces the queueing delay and the loss probability, which both contribute to

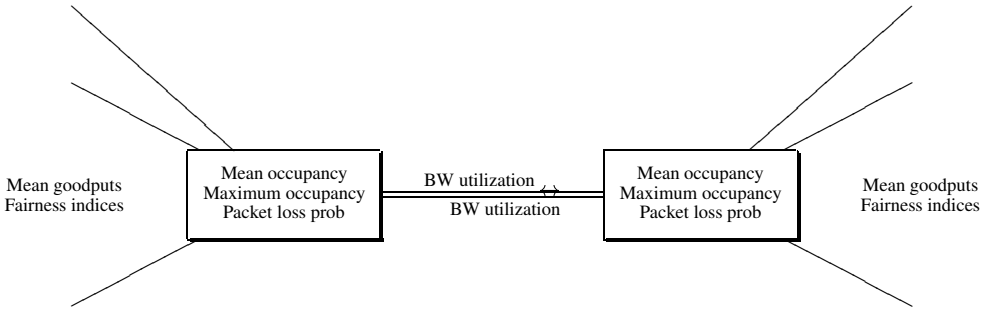

Figure 11: Organization of the results.

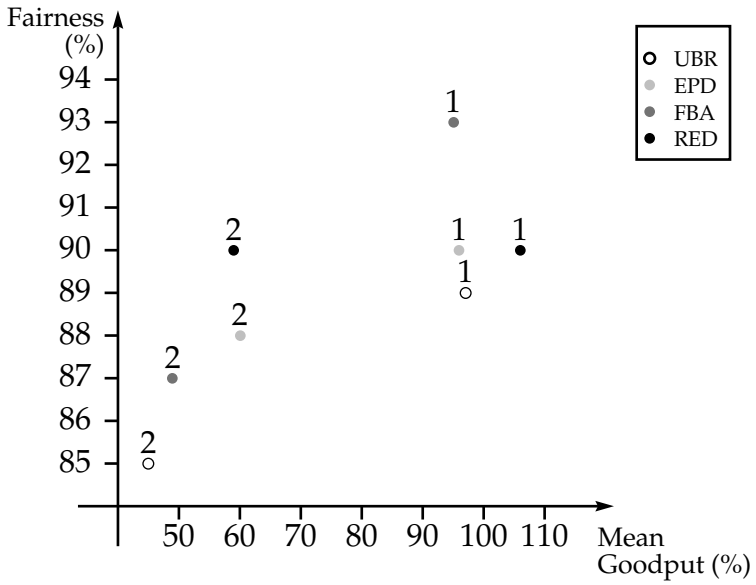

Figure 12: Workstation results: Mean goodput and fairness for the direct access on a single-bottlenecked ATM network. Labels 1 and 2 refer to clusters with 2.5 and 10 ms access delays respectively

increase the TCP throughput, and thereby the TCP goodput. All other methods do not provide any way to monitor the buffer occupancy; more seriously, the maximum buffer occupancy hits the limit for UBR and FBA, which is a strong evidence of buffer overflow. As for EPD, its fixed threshold of 14000 cells (about 90 %) is clearly visualized.

4.2.2 Single-bottlenecked ATM network with a GEO satellite link

Since the only parameter that changes from the corresponding "non-GEO" scenario (section 3.1.1) is the bottleneck delay, the ideal goodput for this environment is the same as the one for the environment without satellite link, namely 13.45 Mbps.

Compared to the previous scenario, one can notice that the larger delays have an expected negative impact on the goodputs which remain around 16 %, but that the fairness is improved. Here, the difference between the short-delayed clusters and the long-delayed cluster fades away, because these delays remain short with respect to the GEO satellite delay (250 ms). The low goodputs with all the discard methods are not due to heavy losses. On the contrary, the packet loss probabilities are very low. The explanation has more to do with the particularly low mean buffer occupancy (around 2 to 4 %), which makes it difficult to keep the line busy.

Overall, FBA is slightly better, but the differences are not really meaningful, and RED turns out to be better as regards the buffer occupation.

4.2.3 Asymmetrical access on a single-bottlenecked ATM network

By definition, the normalized mean goodput *is* the bandwidth utilization in case there is only one cluster in the simulated scenario. This is precisely what arises for the present

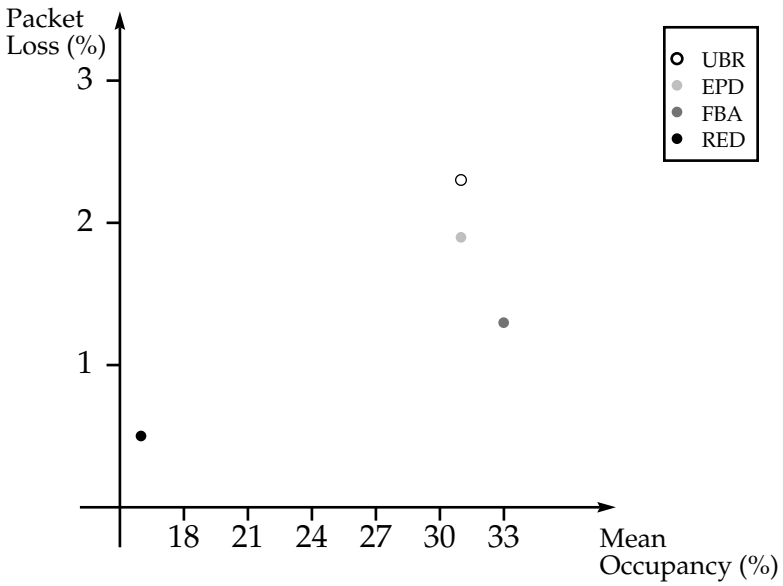

Figure 13: Switch results: Mean occupancy and packet loss for the direct access on a single-bottlenecked ATM network

Table 1: Other results for the direct access on a single-bottlenecked ATM network

	UBR	EPD	FBA	RED
Workstation results				
Global intra-cluster fairness index (%)	87	89	90	90
Link results				
Bandwidth utilization (%)	71	78	72	**82**

environment : as described in section 3.1.2, the workstations are identical (there is no more difference in the access delay). Thus, we do not include "link results" as in other scenarios. The same applies to fairness indices : the general fairness index is equal to the cluster index. This leads to the results as presented in figures 14 and 15 and table 2.

The best overall method is again RED, which exhibits an excellent utilization of the network, a very good fairness between the sources, and a lower buffer occupancy.

4.2.4 Direct access on a GFC-shaped ATM network

As already detailed in section 3.1.3, the scenario is designed to obtain ideal values of goodputs for each cluster of workstations. In this case, with the chosen bandwidths, the expected goodputs for workstations belonging to clusters , , and are 4.04, 1.35, 8.07, 1.35 and 4.04 Mbps respectively.

From figure 16, a general remark is that the fairness between clusters, for example between clusters and on the left link, clusters , and on the central one, and clusters and on the right one. This is an important survey to make in GFC-like environments. This is simply seen by comparing the corresponding mean goodputs, which should normally be as close as possible to their normalized goodput of 100 %. We are far from this picture however.

For UBR and EPD, nothing is really done that could achieve this fairness between clusters. As far as RED is concerned, the loss probability of a VC in a switch is directly related to its average buffer occupancy, but this feature tends to penalize VCs that cross more switches, due to the cumulative effect of random discards. This feature leads to an unfairness towards sources with higher round trip times. FBA should normally perform better, because, the discard method takes the load into account and, at least in steady-state, its deterministic discard is not cumulative. In fact, we will see in section 4.3.2 that with a similar GFC scenario in which ATM is limited to the backbone, FBA performs much better for this criterion than other discard methods. It is not clear why FBA provides no gain in inter-cluster fairness in the present scenario. We conjecture that this is because the packets are larger here (9140 bytes) and thus the TCP fast retransmit works less efficiently. Moreover, when a switch loses several packets in a row on a VC, these packets are from the same workstation, which is not the case when several traffics are aggregated in the same VC. These losses occurring in bursts are likely to generate more timeouts and slow starts, which leads to performance degradations.

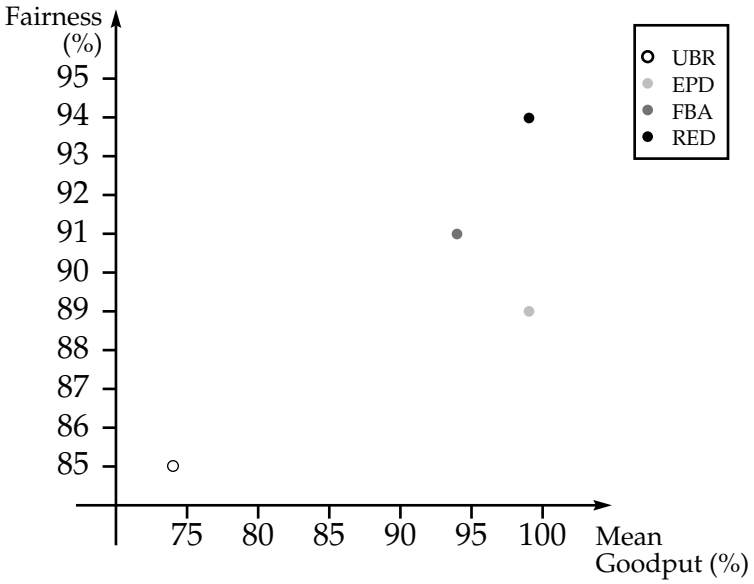

Figure 14: Workstation results: Mean goodput and fairness for the asymmetrical access on a single-bottlenecked ATM network

Table 2: Other results for the asymmetrical access on a single-bottlenecked ATM network

	UBR	EPD	FBA	RED
Switch results				
Maximum occupancy (%)	100	92	100	**82**

As shown in figures 16 and 17 and table 3, despite this general lack of inter-cluster fairness, the best overall method is again RED according to all our criteria : goodputs, fairness among sources in the same cluster, buffer occupancy and link utilization.

Note that a rather surprising result is FBA's global (intra-cluster) fairness index, which is the poorest of the four! We guess that this is because FBA, like UBR, suffers from buffer overflows, as exemplified by the fact that the maximum occupancy hits the 100 % in all the switches. When this occurs, the packet drops are not driven any more by the normalized shares of the buffer computed by FBA, thereby drifting away from the fair shares.

We also compared the performance of the discard methods in a GFC-shaped ATM network with a GEO satellite link or with asymmetrical access. Because of lack of space, we do not provide detailed results, but basically these additional scenarios confirm the results already obtained. Again, none of the discard methods were able to avoid a rather strong unfairness among clusters. In particular, clusters and have goodputs far below those of clusters and respectively. Within clusters, though, EPD and especially RED allow the bandwidth to be more fairly shared.

4.3 IP-based architecture

4.3.1 Router access on a single-bottlenecked ATM backbone

For this first simulation in an IP-based architecture, we can see in figure 18 that the rather expected goodput collapse between the different delayed workstations is not quite so pronounced than for scenario 1 : in the latter, a drop of half the expected goodput could be observed, while in the present case only 10 to 20 % are lost by the 10 ms workstations. Figures 18 and 19 also show that RED is again superior for many performance criteria, such as the buffer occupancy, packet loss and fairness among sources in the same clusters. However, FBA tends to level the differences of goodputs between clusters in this case, which gives a better efficiency while maintaining high goodputs.

4.3.2 Router access on a GFC-shaped ATM backbone

Figure 20 calls for the same important remarks that were made regarding results expressed in figure 16. For the present scenario, we can affirm that FBA is the best overall method. This can be explained as follows : as the general efficiency ("link results") is very high for all strategies as shown in table 5, the fairness between related clusters becomes a more important issue, which is best resolved with

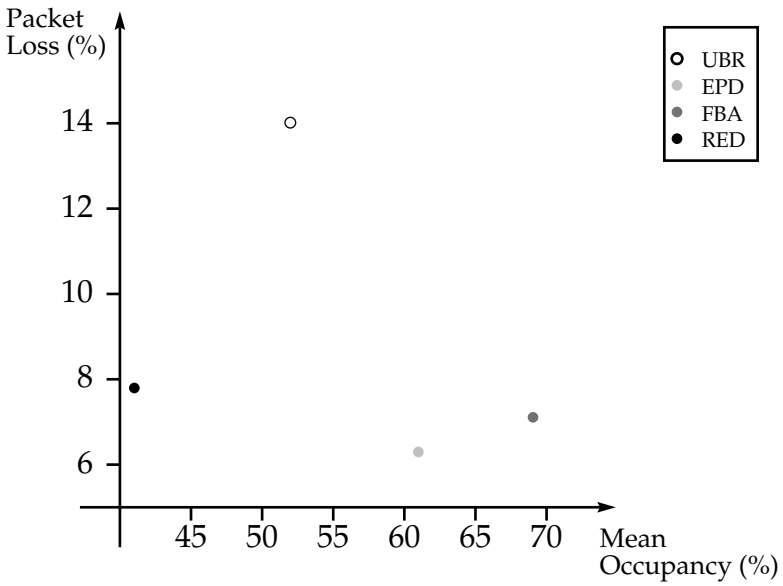

Figure 15: Switch results: Mean occupancy and packet loss for the asymmetrical access on a single-bottlenecked ATM network

Table 3: Other results for the direct access on a GFC-shaped ATM network

	UBR	EPD	FBA	RED
Workstation results				
Global intra-cluster fairness index	88	89	85	**99**
Switch results				
Switch 1 maximum occupancy (%)	100	90	100	**72**
Switch 2 maximum occupancy (%)	100	91	100	**87**
Switch 3 maximum occupancy (%)	100	90	100	**84**
Link results				
Left bandwidth utilization (%)	89	**97**	92	**98**
Middle bandwidth utilization (%)	79	83	81	**92**
Right bandwidth utilization (%)	75	90	78	**99**

FBA. Indeed, the three other strategies allocate the various bandwidths in such a manner that the "one-hop clusters" get way too much resources. Remember that the pairs and *should* get the same bandwidths. Here, for example, a goodput of 34 % and a goodput of 154 % tell us that with plain UBR, the perturbative cluster has taken almost *five* times the bandwidth that has received! This situation can be observed for EPD and, to a lesser extent, for RED as well. On the other hand, FBA exhibits goodput results that are much closer to 100, which is an evidence that the fairness objective between clusters has been reached. Note that FBA is not at all the best method when considering the various intra-cluster fairness indices, where RED gives by far the best results. Finally, figure 21 shows again the superiority of RED as regards the switch results.

5 Conclusions

We have defined a variant of the RED discard strategy for ATM networks. We have assessed its performance and compared it to several other algorithms, namely EPD, FBA, and plain tail-drop UBR. As performance criteria we focused on the TCP goodput and the fairness among TCP connections in several quite different environments.

It is interesting to observe that the results are consistent in the many distinct scenarios, so that we can draw some general conclusions. Firstly, it is reassuring that EPD, FBA and RED give better results than the plain UBR. EPD gives satisfying results in general, especially as regards the switch characteristics (low packet loss probability) and the fairness among sources within the same cluster. However, FBA and RED are almost always superior to EPD. It turns out that FBA performs much better in IP-based scenarios than in end-to-end ATM ones. This is the case for all criteria, but especially for the fairness between clusters, where FBA clearly

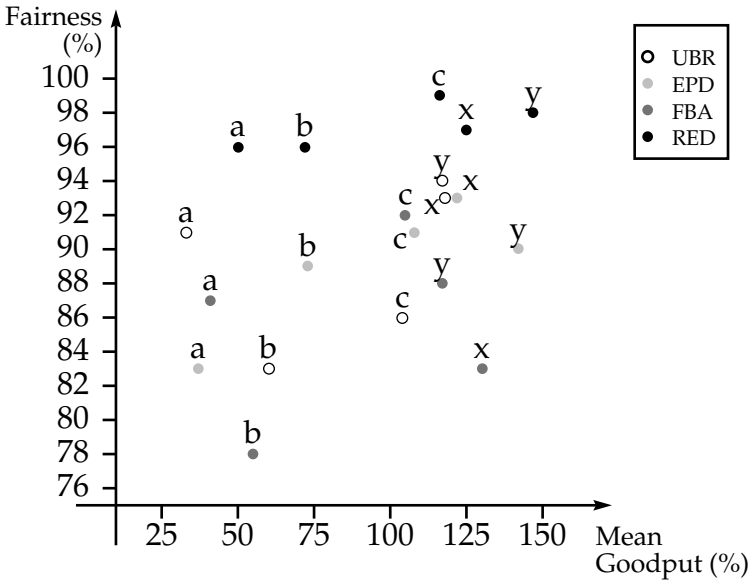

Figure 16: Workstation results: Mean goodput and fairness for the direct access on a GFC-shaped ATM network. Labels a, b, c, x and y refer to corresponding clusters

Table 4: Other results for the direct access on a single-bottlenecked ATM backbone

	UBR	EPD	FBA	RED
Workstation results				
Global intra-cluster fairness index (%)	78	78	79	**85**
Switch results				
Maximum occupancy (%)	100	**88**	100	**88**
Link results				
Bandwidth utilization (%)	78	86	**90**	86

outperforms RED. It is not clear why FBA is less effective in end-to-end ATM scenarios, but this could be due to a poor parameter tuning, as exemplified by a maximum buffer occupancy reaching 100 %. Conversely, RED has by far the lowest mean buffer occupancy in general, which gives low delays, while offering high goodputs and link utilizations. A mean buffer occupancy around RED is also a good solution as regards the fairness among the similar sources in the same cluster, especially in IP-based scenarios, but is poor at achieving fairness between different clusters having different characteristics. In particular, sources with higher round trip times, or crossing more hops, have lower goodputs.

6 Acknowledgements

This work was partially supported by the Flemish Institute for the promotion of Scientific and Technological Research in the Industry (IWT). We are very grateful to Roch Guerin, the editor, and to the anonymous reviewers for their useful and insightful comments, and to Ludovic Kuty for his drawings of the simulations results.

References

[BCC+98] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and congestion avoidance. Internet RFC 2309, April 1998.

[CT97] M. Casoni and J. Turner On the performance of Early Packet Discard. *IEEE Journal on Selected Areas in Communications*, 15(5):892–902, June 1997.

[EA97] O. Elloumi and H. Afifi RED algorithm in ATM networks. In *IEEE ATM'97*, Lisboa, Portugal, May 1997.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection for congestion avoidance. *IEEE/ACM Transactions On Networking*, 1(4):397–413, August 1993.

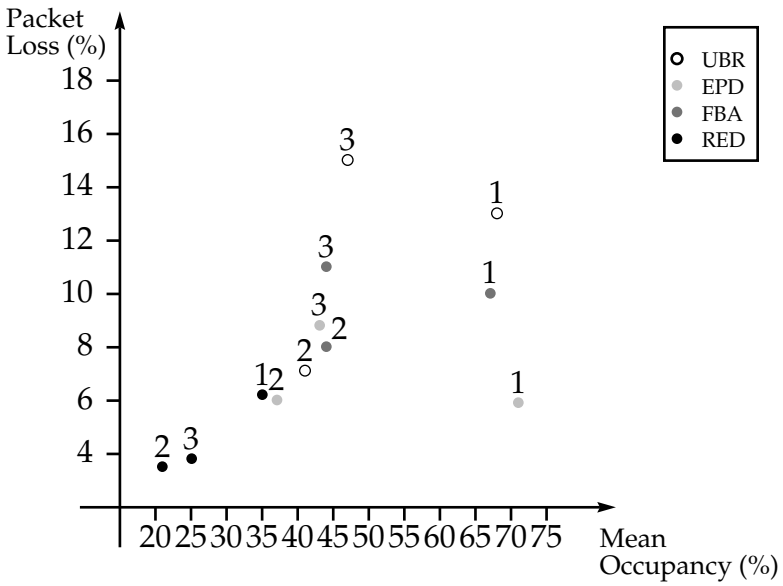

Figure 17: Switch results: Mean occupancy and packet loss for the direct access on a GFC-shaped ATM network. Labels 1, 2 and 3 refer to the three switches from left to right

Table 5: Other results for the direct access on a GFC-shaped ATM backbone

	UBR	EPD	FBA	RED
Workstation results				
Global intra-cluster fairness index (%)	75	81	78	**99**
Switch results				
Switch 1 maximum occupancy (%)	100	88	100	**47**
Switch 2 maximum occupancy (%)	100	88	97	**42**
Switch 3 maximum occupancy (%)	100	88	100	**43**
Link results				
Left bandwidth utilization (%)	93	97	98	98
Middle bandwidth utilization (%)	94	98	97	98
Right bandwidth utilization (%)	94	99	98	99

[Flo97] S. Floyd. Optimum functions for computing the drop probability. Email available at `http://www-nrg.ee.lbl.gov/floyd/REDfunc.txt`, October 1997.

[GJKF98] R. Goyal, R. Jain, S. Kalyanaraman, and S. Fahmy. Improving performance of TCP over ATM-UBR service. *Computer Communications*, 21(3):898–911, 1998.

[HK98] J. Heinanen and K. Kilkki. A fair buffer allocation scheme. *Computer Communications*, 21:220–226, 1998.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOMM88*, pages 314–329, August 1988.

[KKTO97] K. Kawahara, K. Kitajima, T. Takine, and Y. Oie. Packet loss performance of selective cell discard schemes in ATM switches. *IEEE Journal on Selected Areas in Communications*, 15(5):903–913, June 1997.

[LM97] D. Lin and R. Morris. Dynamics of Random Early Detection. In *SIGCOMM 97*, pages 137–145, Cannes, France, September 1997.

[LNO96] T. Lakshman, A. Neidhardt, and T. Ott. The drop front strategy in TCP and in TCP over ATM. In *Proceedings INFOCOM96*, pages 1242–1250, 1996.

[Mah96] J. Mahdavi. Experimental TCP selective acknowledgment implementation. Available from `http://www.psc.edu/networking/tcp.html`, 1996.

[Man96] S. Manthorpe. STCP 3.2.6. Available from

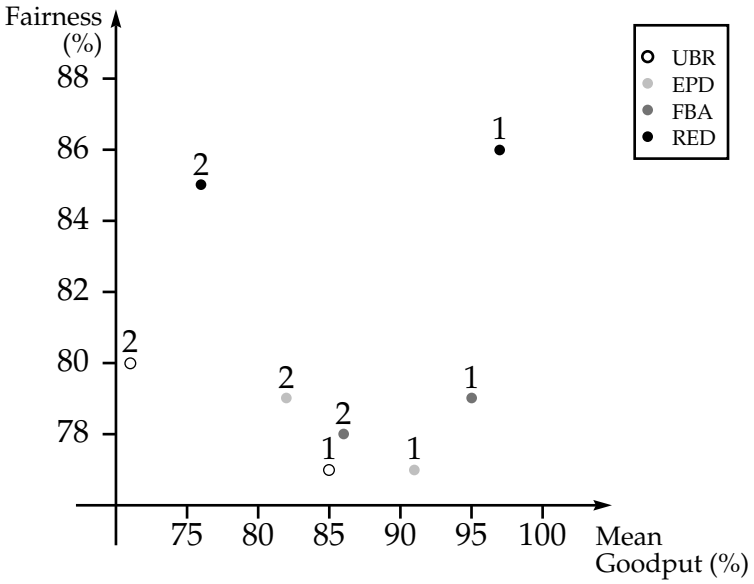

Figure 18: Workstation results: Mean goodput and fairness for the direct access on a single-bottlenecked ATM backbone. Labels 1 and 2 refer to clusters with 2.5 and 10 ms access delays respectively

`http://lrcwww.epfl.ch/~manthorp/stcp/`, 1996.

[RBL98] V. Rosolen, O. Bonaventure, and G. Leduc. Impact of cell discard strategies on TCP/IP in ATM UBR networks. In *IFIP ATM'98 Workshop*, Ilkley, UK, July 1998.

[RF95] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(4):633–641, May 1995.

[RF99] K. Ramakrishnan and S. Floyd. A proposal to add Explicit Congestion Notification (ECN) to IP. Internet RFC 2481, January 1999.

[Sim94] R. Simcoe. Test configurations for fairness and other tests. ATM Forum contribution 94-0557, July 1994.

[TMW97] K. Thompson, G. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network Magazine*, 11(6), November/December 1997. Also available from `http://www.vbns.net/presentations/papers`.

[Tur96] J. Turner. Maintaining high throughput during overload in ATM switches. In *INFOCOM 96*, March 1996.

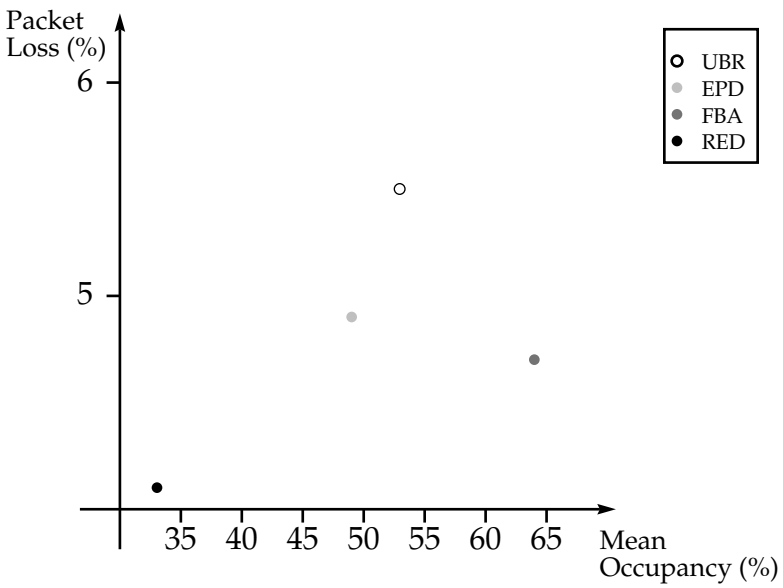

Figure 19: Switch results: Mean occupancy and packet loss for the direct access on a single-bottlenecked ATM backbone

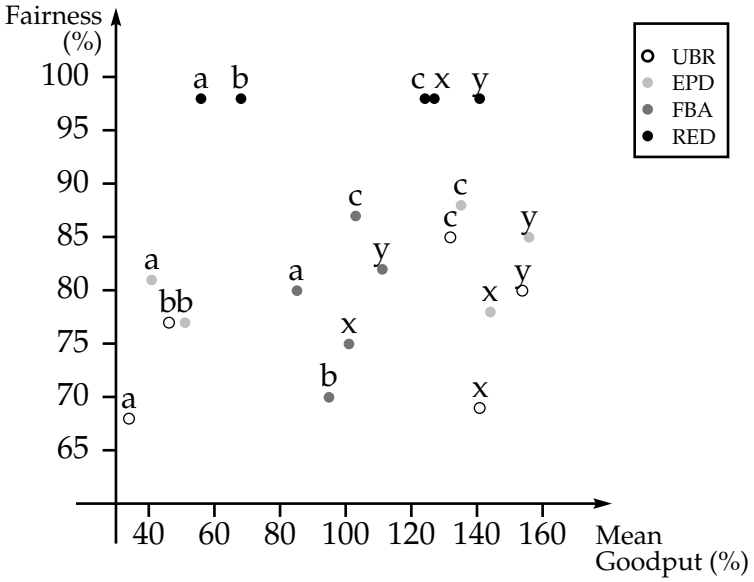

Figure 20: Workstation results: Mean goodput and fairness for the direct access on a GFC-shaped ATM backbone. Labels a, b, c, x and y refer to the corresponding clusters

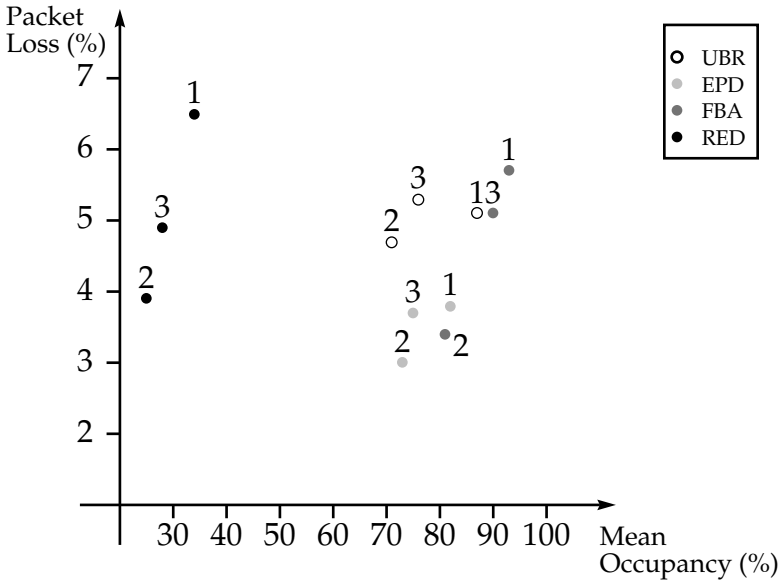

Figure 21: Switch results: Mean occupancy and packet loss for the direct access on a GFC-shaped ATM backbone. Labels 1, 2 and 3 refer to the three switches from left to right