Simulation Study of PRCA and FCVC in ATM LANs Supporting TCP/IP*

Mario Gerla, Carlos Pazos
UCLA-Computer Science Departement
Los Angeles, CA 90024

Vincenzo A. Signore
Politecnico di Bari
Italy

Abstract

TCP/IP traffic is likely to be one of the most heavily carried over ATM networks. However this kind of traffic is of connectionless nature, while the ATM network is connection-oriented. For connectionless traffic it is not possible for the user to specify the values of the traffic descriptors required by the ATM network for resources reservation. For this reason connectionless traffic will be carried on an Available Bit Rate (ABR) basis, allowing it to fill in the bandwidth left over by services for which resources have been allocated, but with no performance guarantees.

In order to prevent the ABR traffic from congesting the network, special conegestion control techniques have been devised for this purpose. These schemes can be categorized in two major groups: rate-based and credit-based schemes. The most important representatives of these categories are respectively the Proportional Rate Control Algorithm (PRCA) and the Flow Controlled Virtual Circuit (FCVC). Since TCP/IP traffic is also controlled by the TCP window flow control, in this paper we present a comparison between PRCA and FCVC under TCP traffic. In our study we simulate an ATM LAN in which TCP and guaranteed performance sources share a common congested link. We intend to determine which of the considered schemes is most successful in allowing the TCP traffic to be used as fill-in traffic. In this sense the performance measure we are mostly concerned with is the goodput, namely the effective throughput seen by the applications

*This work was supported by NSF through grants NSF 9305376 and NSF-INT 9115882.

on top of TCP. The goodput is measured with different levels of congestion, simulated by varying the burstiness for the guaranteed sources and the number of active TCP connections. We also investigate the impact of the maximum size of TCP segments, which seems to be critical for the PRCA approach, since cells can still be lost with this scheme.

1 Introduction

During the last few years the networks evolution has been driven by two main factors: the proliferation of Internet applications, mostly supported by the TCP/IP protocol stack, and the gradual maturation of ATM technology both in the local and in the wide area scenario. It is likely that the next generation of LAN will be based on an ATM subnetwork and support TCP/IP protocols, in order to guarantee the compatibility with existing LAN services. Moreover the possibility of efficiently integrating different classes of services will push the development of new multimedia applications. Hence, the problem facing us is how to carry the TCP/IP connectionless traffic on an ATM network, without disrupting the other, possibly multimedia, connection oriented traffic. Namely, since IP is a connectionless protocol, while ATM is a connection oriented network, the ATM resources negotiation phase at call setup time is unfeasible for TCP/IP connections. ATM interfaces supporting TCP/IP based applications are unable to declare their own traffic parameters, thus no QoS can be guaranteed. As a result, such services are to be included in the ABR class, meaning that they do not have reserved resources, but they are allowed to use the bandwidth left available by high-priority CBR and VBR sources along the path toward the destination. Given this scenario, it is of great importance to understand how ATM congestion control schemes for ABR services are affected by the TCP window flow control. In this paper credit-based and rate-based schemes are compared via simulation of an ATM LAN, with ATM workstations running a sample TCP/IP application and background traffic generated by VBR sources.

2 Congestion control schemes for ABR services

A congestion control algorithm for ABR services in ATM networks should be the result of a trade-off among these main factors:

- *Effectiveness.* ABR sources must be allowed to aggressively exploit the bandwidth not allocated to services with guaranteed performances, without affecting their QOS. A good measure of this property is the link utilization.

- *Responsiveness.* Whenever congestion builds up, best-effort traffic must be promptly slowed down in order to prevent switch buffers overflow. A responsive control algorithm has to keep as low as possible the cell loss probability for ABR services, although a maximum value can not be guaranteed.

- *Low impact on switch architecture.* The algorithm must be simple enough to be implemented efficiently without dramatically increasing the switch complexity and cost.

- *Fairness.* The reactive action, undertaken by the control algorithm in case of congestion, should equally affect ABR users, belonging to the same priority class.

Most of the congestion control methods proposed for best-effort traffic in ATM networks are rate-based, meaning that the transmission rate is adjusted based on information about the level of congestion within the network.

Bandwidth advertising schemes [GTG93] fall in this category. ABR users are connected through "thin" semi-permanent connections, without allocated resources. Periodically each source is notified with the bandwidth available along its connections, which can be entirely or partially used. High link utilization is achieved, but the responsiveness is low, because there is no control on the aggregate traffic and many sources can compete for the residual bandwdith on common links. The impact on switch complexity is limited.

Other rate-based schemes are based on Explicit Congestion Notification (ECN) [New93]. Switch buffer lengths are constantly monitored and if a threshold is exceeded cells are marked. With the Forward ECN, both marked and unmarked cells are forwarded to the destination, which counts them. Depending on the ratio between the counters, a management cell with either congestion or no congestion indication is sent to the source. The source reacts to such a notification, either by multiplicatively decreasing or additively increasing its rate. The responsiveness of this scheme might be penalized by long round-trip propagation delays. Fast reaction to congestion can be

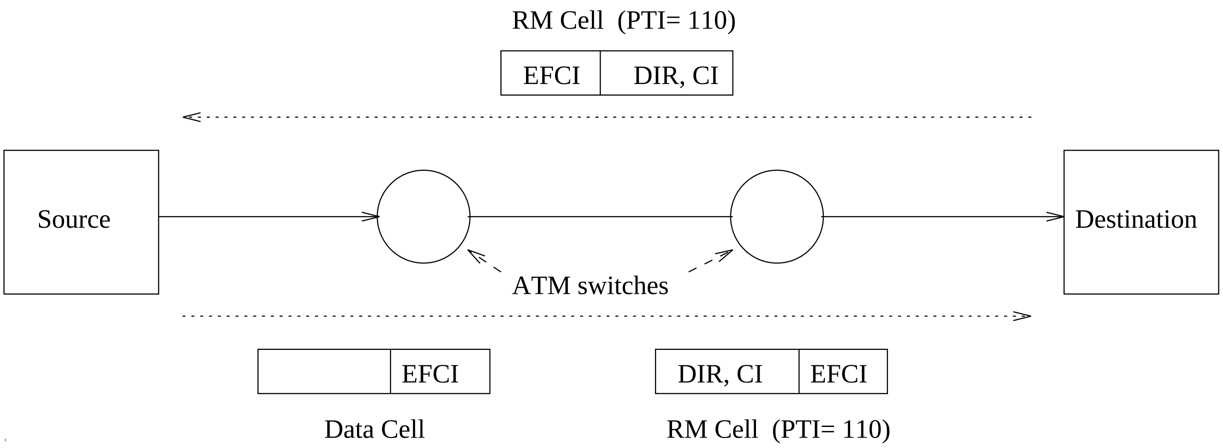

Figure 1: Representation of the closed-loop control with the PRCA.

achieved by the Backward ECN, which charges each switch with the responsability of notifying the sources. Obviously, the complexity of the switch design can be prohibitive and if many links along a VC path are congested, the source is likely to be flooded with congestion indication messages.

Recently, the two congestion control schemes which have most attracted the attention of the ATM Forum are the PRCA (Proportional Rate Control Algorithm) [ea94], which is rate-based, and FCVC (Flow Controlled Virtual Circuit) [KC94], a credit based scheme. In FCVC each VC has a buffer allocated at each crossed ATM switch, and the transmission of a cell through an ATM link is allowed only if the credit balance, reflecting the amount of buffer available in the upstream switch, is positive. A detailed description of these two algorithms follows.

2.1 Proportional Rate Control Algorithm

The basic idea is that the source is allowed to increase its rate only when it receives an explicit indication of no congestion, otherwise it keeps additively decreasing the rate after every cell transmission, down to the minimum value MCR (Minimum Cell Rate). The decrease step is determined by the parameter ADR (Additive Decrease Rate), depending on a negotiated Multiplicative Decrease Factor (MDF). The network is assumed to be segmented into domains each delimited by virtual or actual source-destination couples. The domain end points run a closed management loop: every cell is sent with the Explicit Forward Indication (EFCI) bit set to zero, and eventually will be marked by one of the intermediate switches in case of congestion. See fig. 1 for a schematic description of PRCA closed-loop control.

Each domain chooses its way to detect congestion. One of the traditional ways is monitoring the buffer length and marking cells when a predefined threshold is exceeded. After a given number of data cells, specified by the parameter NRM, are transmitted, a Resource Management cell is sent by the source in order to probe the domain. A Congestion Indication (CI) bit is provided in each RM cell. This bit is set by the destination if the last data cell was received marked, or left zero otherwise, and the RM cell is sent back to the source. Upon receiving a RM cell with no congestion indication, the source is allowed not only to compensate for the rate decrease in the last cycle, but also to increase it further by an agreed upon Additive Increase Rate (AIR), without exceeding a negotiated Peak Cell Rate (PCR). If the congestion bit is set, the source continues reducing the rate. Of course, if a connection spans many domains, the virtual source of each domain is in charge of closing the loop for the previous domain, while the virtual destination starts the loop in the next one. An enhanced version of this algorithm, enriched with a bandwidth advertising mechanism, is described in [ea94].

2.2 Flow Controlled Virtual Circuit

A schematic description of FCVC link-by-link credit control is presented in fig. 2. The algorithm is conceptually straightforward. Each node (including the destination interface) has an allocated buffer for every crossing VC. The size of a single buffer is $N_2 + N_3$, where: N_2 is the amount of cells the node has to forward before sending a credit cell back to the previous node along the considered VC; N_3 is the amount of cells that could arrive within a link Round-Trip Time RTT at a targeted bandwdith B_{VC}. If N_{bits} is the number of bits in an ATM cells ($53 * 8 = 424$ bits/cell):

$$N_3 = \left\lceil \frac{RTT * B_{VC}}{N_{bits}} \right\rceil \tag{1}$$

The node keeps a credit count for each VC, which decreases by one every time a cell is transmitted. Of course a cell transmission is possible only when the credit count is non null. Moreover it keeps a count C of the cells sent through each VC. This value is inserted in an apposite field of the credit cell, when sent back to the upstream node. Upon receiving a credit cell the upstream node

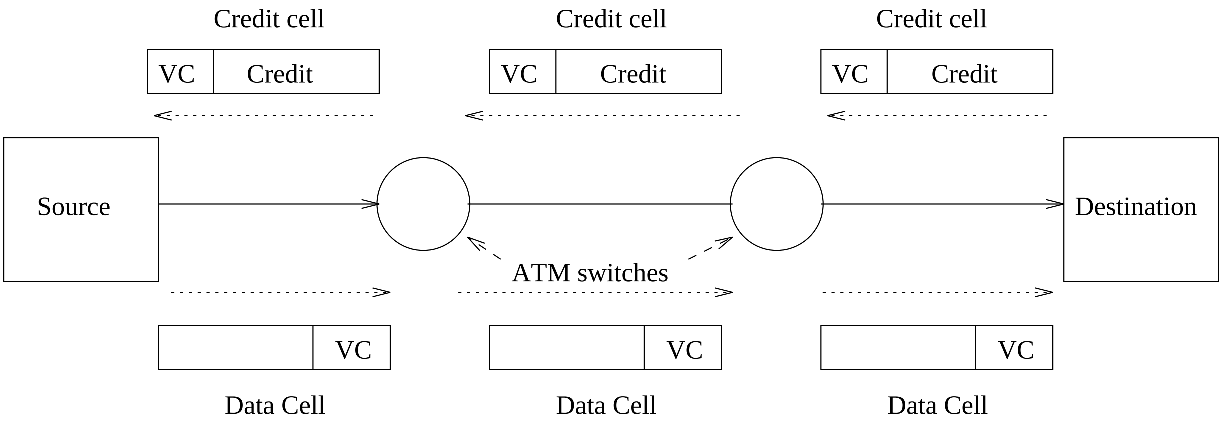

Figure 2: Representation of the credit-based control with FCVC.

updates its own credit count summing $C_{new} - C_{old}$ to the current value, where C_{new} and C_{old} are the respective values of C when the current and last credit cells are sent. An enhanced version of this scheme, featuring an adpative buffer allocation technique, is described in [KC95]

2.3 FCVC versus PRCA: implementation issues

Although the link-by-link flow control appears conceptually simple, the implementation complexity is significant. First of all, control functions are required node by node, whereas ATM was conceived according to the "core-and-edge" principle: control functions are to be implemented at the edges of the network (UNI), but not inside it (NNI) in order to keep the protocols as simple as possible, so to be able to switch small size cells at high speeds. Furthermore each node needs to keep track of the amount of cells transmitted through every VC crossing it, in order to let the upstream node update its credit balance upon receiving the credit cell. An alternative would be to fill the credit field with the amount of space available in the VC buffer, but this solution does not allow to recover from temporary credit starvations caused by lost credit cells. Memory management is probably the most negative aspect of the original FCVC. The switch controller has to handle independently several separated buffers, which have in general different sizes in an heterogeneous traffic environment. The amount of memory is proportional to the link propagation delays leading to very large buffers requirements in the wide area scenario and no scalability of the switch design. Finally, the static buffer allocation is probably wasteful for high burstiness traffic. The modified algorithm [KC95] partially alleviates some of these problems. In conclusion the implementation seems to be feasible only in ATM LANs with short propagation delays and a reasonably low number of end-to-end VCs.

On the other hand PRCA has a far lower impact on the ATM switch architecture, the only problem being the parameter tuning at connection setup time. Indeed most of the parameters (PCR, AIR, ADR) are meant to be negotiated and defined during the signalling procedure, even though for connectionless best-effort traffic not enough information can be specified a priori by the connection originator. Anyway, the same problem may appear with FCVC for the definition of the VC targeted bandwidth.

2.4 FCVC versus PRCA: interaction with TCP

FCVC guarantees no cell losses, unless the source is misbehaving and ignores a zero credit count. Moreover, low priority best-effort cells are backpressured in case of congestion and used to fill in the "holes" in the transmission stream of high-priority guaranteed traffic. Thus, FCVC is expected to show high responsiveness and effectiveness.

PRCA does not insure zero losses and its responsiveness is dependent on the propagation delays of RM cells and on the effectiveness of the monitoring algorithm inside the switch. Effective bandwdith fill-in can be achieved if the rate adjustment algorithm quickly converges to a fair share of the residual bandwidth. Again, the shorter are the propagation delays the faster is the convergence. A trade-off between responsiveness and effectiveness can be carried out, by appropriately tuning the parameters MDF and AIR. However, with PRCA the link utilization itself might not be a meaningful measure of effectiveness. Indeed, if TCP/IP is supported, cell losses trigger TCP segment retransmission, and a portion of bandwdith is wasted along the path for unsuccessful transmissions. Moreover, since many TCP sources are likely to be simultaneously affected by cell losses in case of congestion, deep throughput ripples can occur if synchronized retransmission are not avoided. For this reason defining an appropriate TCP Maximum Segment Size (MSS) can be a critical issue with PRCA. With small segments the TCP/IP header overhead can be remarkable, but increasing the segment size can be harmful because of the need of retransmitting large amounts

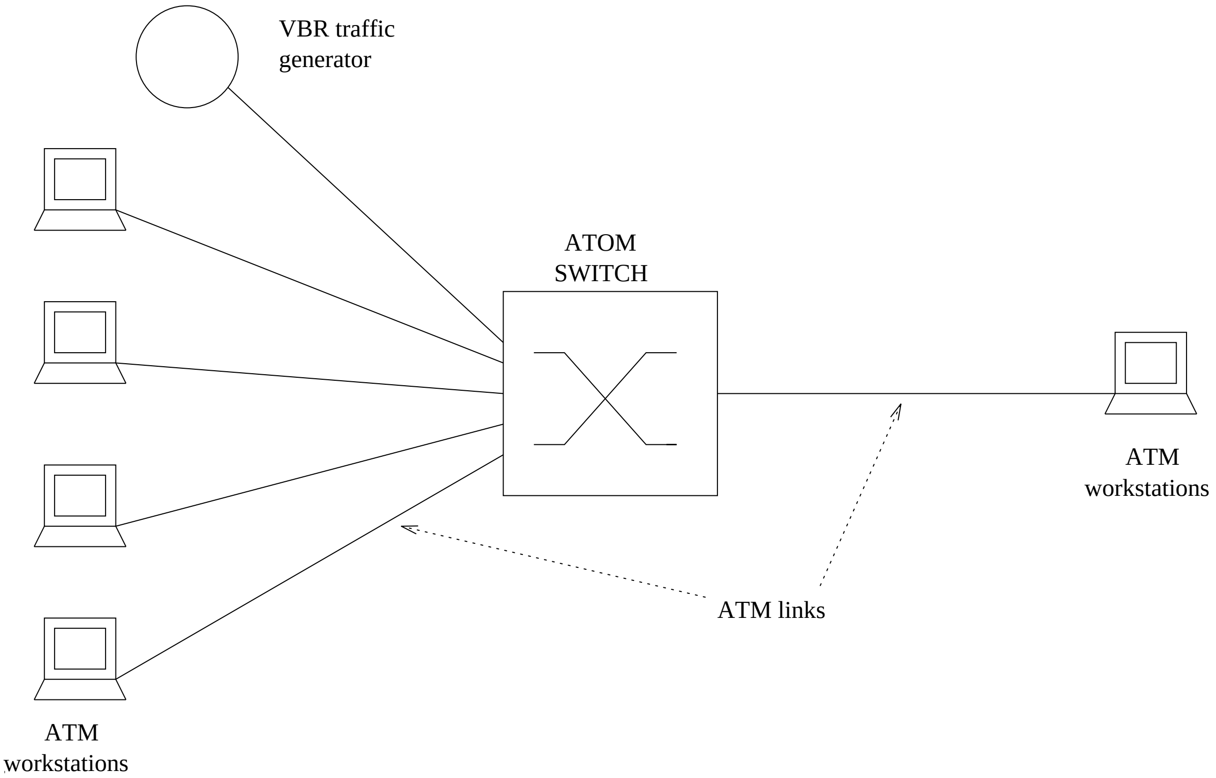

Figure 3: Scheme of the ATM LAN model.

of data. This problem is expected to lead to a significant waste for a network supporting large file transfer applications, especially in case of sudden and short periods of congestion. On the contrary FCVC can take more advantage of an increase of the MSS, because TCP retransmissions are in general not required.

In the following, FCVC and PRCA are compared through the simulation of an ATM LAN supporting TCP/IP. The main performance measure is the goodput, that is the throughput as seen by the application on top of TCP.

3 Simulation of an ATM LAN: assumptions

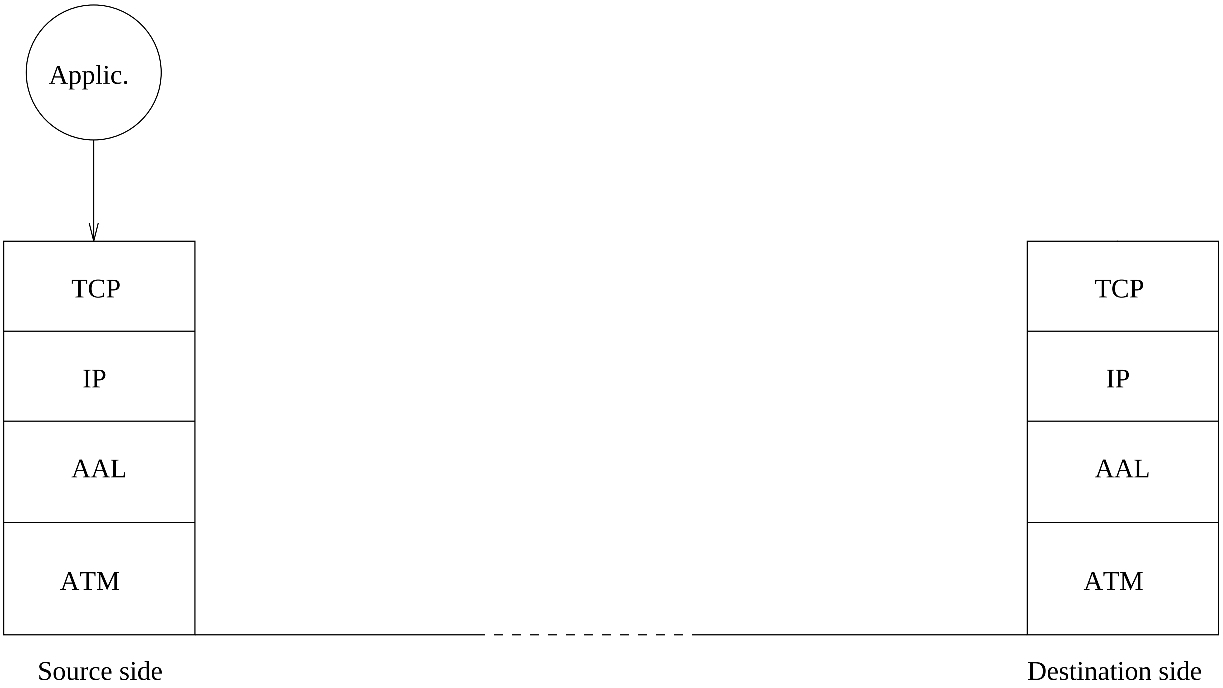

Figure 4: Layered protocol structure of the ATM workstations.

The model of the ATM LAN used in our simulation program, written in MAISIE language [BL90], is based on a single switch (fig. 3). We use the ATOM [SNS+89] architecture as a model for the basic switching functions and specific features are added at ATM and AAL levels in order to support either PRCA and FCVC congestion controls. All traffic sources share a single outgoing link, where bandwidth is allocated only to guaranteed sources, while best-effort sources are competing for the residual bandwidth. A more detailed description of each element of the simulator follows.

3.1 Guaranteed traffic generator

For the sake of simplicity we assume the guaranteed traffic comes through a single input link. It is generated as an MMDP process resulting from aggregating VBR sources. Each source has the following characteristics:

- peak rate $B_{pg} = 10\ Mbit/s$ at the user level;

- mean burst length $L_g = 4.8\ Kbytes$;
- variable burstiness b_g.

The number of sources $N_g = 14$ has been chosen such that, when they are all simultaneously active, no residual bandwidth is left on the common link. By varying the value of b_g, it is possible to get the desired average bandwidth B_{ag}. The instant value of the bandwidth available to TCP sources varies as time goes by, because of the variations of the number of simultaneously active guraranteed sources. By doing so we can study which scheme allows for a better fill in.

3.2 ATM workstations

A layered protocol structure of the model used for a single ATM workstation is depicted in fig. 4.

The client application is represented through an ON-OFF model with exponential active and idle periods. The main traffic parameters are:

- average message length $L = 2.5\ Kbytes$;
- average bandwidth $B_a = 2.5\ Mbit/s$, at the user level;

The number N_{tcp} of applications, and consequently of required TCP connections, is varied from 1 to 15, such as to simulate different levels of congestion. All TCP connections are kept opened throughout the whole simulation (20 seconds of actual time). Recent measures on actual TCP traffic have shown [PF94] that the aggregate traffic shows a long-term correlation not taken into account by Poissonian models. Arrivals within a connections are better represented by heavy-tailed distributions. However, for small time scales the arrival process is dependent on the network dynamics and it is still not enough clear how to model an individual source at the *application* level, where resource sharing and lower level protocols have no effect on the generated traffic. For this reason we still use at the application level a classical Poissonian model.

The TCP level at the source site performs window based flow control functions as described in [Jac88]. The application byte stream is segmented in TCP segments of up to MSS bytes, with a 20 bytes header. Each successfully transmitted segment is acknowledged by the destination side TCP protocol. Upon receiving a positive acknowledgment the source TCP increases the window size by MSS if it is in the slow-start phase, whereas in the congestion avoidance phase it is increased only when all the segments in the current window have been acknowledged. In case of a segment timer expiration, the window size is reduced to MSS. Timer are set based on round trip estimations according to Karn's algorithm [Com91]. In our simulation, timers and round-trip estimations do not include transmission times, meaning that the reference instant is the time of a segment transmission completion. Although this choice may not be feasible for real TCP implementations, it makes the round-trip estimations independent of the segment sizes, allowing for a better representation of the delay, and consequently of the level of congestion inside the network.

For our purposes the IP level performs minor functions. A 20 bytes header is added to each TCP segment.

The AAL level is based on the AAL5 protocol. An 8 bytes CS-PDU trailer is added to each IP packet. AAL and ATM levels perform also operations related to the specific congestion control scheme. The reverse direction is assumed to be uncongested, thus no credit cell, RM cell, or TCP acknowledgments are lost in our simulation. We are mainly concerned with congestion *inside* the network, therefore TCP segments are assumed not to be lost either at the source and destination side.

3.3 ATOM switch

The simulation of the ATOM switch is based on three MAISIE entities as described in fig. 5. The Input Packet Processor performs ATM cell header processing and tags cells for internal routing. The Broadcast Bus routes cells at speed N times the link speed (up to N cells can be switched per time unit). Cells are queued in the appropriate output buffer, and scheduled for trasmission by the Output Packet Processor. For both FCVC and PRCA a double priority buffering is exploited. ABR cells are taken from the apposite buffer only when the guaranteed cells buffer is empty. Since all guaranteed cells are assumed to come through a single incoming link, a single cell buffer is

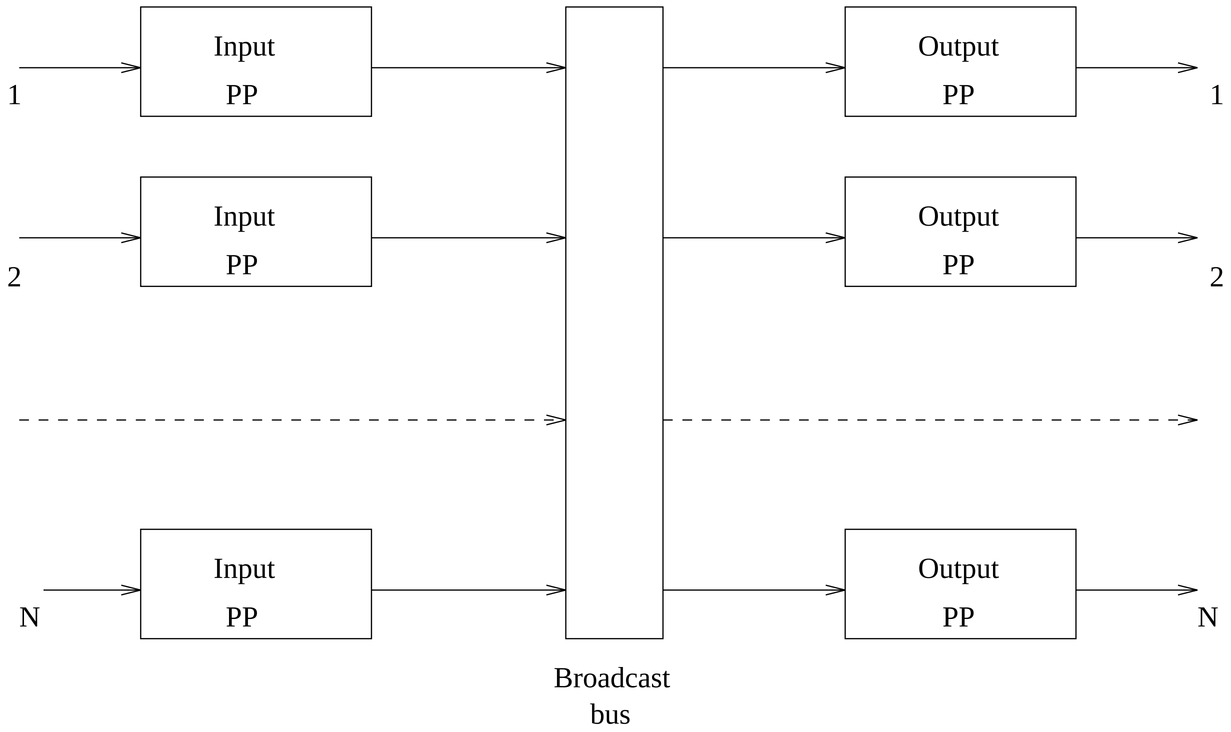

Figure 5: Basic elements of the simulated ATOM switch.

enough. The ABR buffer size is 165 cells. The link propagation delay, including processing time, is 70 μs. This is a reasonable value for a campus-wide network.

3.4 Protocol specific assumptions: FCVC

With FCVC the buffer is divided into up to 15 VC buffers of 11 cells each. The VC buffer size allows to store $N_3 = 1$ cells with the link propagation delay of 70 μs at a targeted user bandwidth of $B_{VC} = 2.5\ Mbit/s$ (see eq. 1), and to send a credit cell every $N_2 = 10$ data cells sent in the downstream direction. Best effort cells are scheduled for transmission according to a simple round-robin policy among all the active VCs.

3.5 Protocol specific assumptions: PRCA

With PRCA the buffer is shared by all best-effort VCs. The congestion threshold is set to 50% of the buffer size. The parameters are chosen such as to sustain a user rate of 2.5 Mbit/s when the network is not congested (all RM cells carry a no congestion indication). In detail: $PCR = 12\ Mbit/s$, $MCR = 500\ Kbit/s$, $ADR = ACR/2^{MDF}$ (when a positive RM cell is received) and $MDF = 5$, $NRM = 32$, $AIR = 3.2\ Mbit/s$.

4 Simulation results

In this section the simulation results are discussed. Our major concern is the effectiveness of the ABR schemes under comparison, expressed by means of the TCP sources goodput. However, results regarding responsiveness end fairness are also reported.

4.1 Effectiveness

A first set of results is presented from fig. 6 to fig. 9. Those curves show the goodput γ as a function of the number of TCP sources for different values of the guaranteed sources burstiness and consequently of the bandwidth available on the common link. We use the values $b_g = 1.1,\ 1.2,\ 1.3$, typical for almost CBR applications, and the TCP maximum segment size is $MSS = 1024\ bytes$. At ATM level the corresponding values of the average available bandwidth are $B_{av} = 13.6,\ 25,\ 34.6\ Mbit/s$ respectively. Fig. 6 and 7 represent the total goodput the TCP applications achieve with PRCA and FCVC respectively. Both plots report also the total traffic the TCP applications generate. It is evident that all curves can be divided in an uncongested region, where the goodput follows the offered traffic, and a congested region, where the goodput reaches a "saturation" value. This saturation level corresponds to the situation of a fully utilized link and as expected increases with b_g. However it is clear that all those saturation levels do not match the corresponding values of available bandwidth. Indeed PRCA achieves approximately $\gamma = 7.3,\ 12.2,\ 16.6\ Mbit/s$, for the respective values of b_g, whereas FCVC gets $\gamma = 9.6,\ 18.9,\ 27.4\ Mbit/s$. Part of the bandwidth available at ATM level is unusable by the

TCP applications because of the intermediate protocols overhead and retransmissions triggered by sudden increases in the round-trip delay, even without actual losses. Moreover with PRCA, ATM cells can be lost and a need arises for segment retransmissions. This extra burden explains the gap of performances between PRCA and FCVC. Fig. 8 and 9 present the average goodput experienced by a single source, compared with the offered traffic. Again we can see that the uncongested region is definitely wider for FCVC than PRCA for all values of b_g. Later on, the issue of how close is the goodput of each individual station to the average values reported in Fig. 8 and 9 is addressed.

As we pointed out above, the limitation in the effective throughput of TCP applications is caused partly by the underlying protocols overhead. Fig. 10 and 11 show the total goodput as a function of the number of TCP sources for $b_g = 1.3$ and for $MSS = 256, 512, 1024, 2048, 4096\ bytes$. It is clear that the goodput increases with MSS for FCVC because the protocol overhead is reduced. However, it is not worth increasing MSS to values much larger than the average message length L because most of the messages are not large enough to fill a single segment, thus there is no advantage. Indeed the curves for the largest MSS values are very close to each other.

With PRCA the situation is different. Two effects are worth to be pointed out. Initially, the advantage in increasing MSS is clear for two reasons. First, the protocol overhead is reduced. Second, very short segments are more prone to retransmissions triggered by increased delays during periods of full activity of the guaranteed sources. This effect does not allow the network to follow the offered traffic, even when only a single TCP connection is active (fig. 12). Larger segments do not suffer from those wasteful retransmissions, because their trasmission time is comparable to or even larger than the periods of high activity of the guaranteed sources. Therefore it is less likely that there are one or more of these segments outstanding, i.e., blocked in the switch buffer and waiting to be transmitted, for which timers can expire. For FCVC (fig. 13), it is unlikely to have more than one outstanding segments because, even with the shortest MSS, the segments are likely to be blocked at the source site when the guranteed sources are active, due to the small sizes of each individual VC buffer. Of course the picture would be different if the transmission times were to be included in the TCP timers and round-trip samples.

Although larger segment sizes reduce the protocol overhead, with PRCA, cells losses trigger TCP retransmissions, and the retransmission of very large segments is wasteful as well. For instance, with $MSS = 4096\ bytes$ the goodput starts to decrease with the number of TCP connections, and reaches values even smaller than for shorter MSS, despite of a very small protocol overhead. This consideration is stressed in fig. 14. The goodput is plotted as a function of MSS, with 15 active TCP connections for both FCVC and PRCA. While for FCVC the goodput is an increasing function of MSS, with PRCA it reaches a maximum around $MSS = 2048\ bytes$, and decreases again for larger MSS. The gap in goodput is caused by the fact that, unlike FCVC, for PRCA there is no possibility of promptly stopping the ABR sources when there is a sudden burst from high priority connections. As a result, TCP connections can be simultaneously affected by cells losses and congestion indications. The former cause the window to shrink to a single segment, while the latter prevent the ATM rate from increasing. Furthermore there might be simultaneous retransmissions which can cause congestion to occur again, if the guaranteed sources are still sending a high volume of cells. The consequence is a fluctuating behavior not only of the goodput but even of the link utilization as shown in fig. 15. The horizontal axis is the the sampling interval index (each interval is 1000 cell transmission slots long). Within the window of time plotted the utilization trace is oscillating for PRCA, though the traffic offered by the TCP connections is enough to fill the residual bandwdith, while FCVC achieves full utilization.

The TCP goodput achieved by FCVC is very close to a theoretical bound, plotted in fig. 14 by considering the average residual bandwidth B_{av} and the fraction η_{OH} of ATM bandwidth available to the application. This fraction, limited by the protocols overhead, is computed analytically in the following as a function of MSS and the distribution of the length of the application messages. For each TCP segments a total of 48 bytes, including TCP and IP headers and a AAL-CS trailer, is added. The CS-PDU is then segmented in 48 bytes payloads and finally the 5 bytes ATM headers are added. Two integer-valued functions are defined:

$$
\begin{aligned}
N_{seg}(x) &= n \quad \textit{if} \quad (n-1)*MSS < x \le n*MSS \\
N_{cells}(x) &= m \quad \textit{if} \quad 48*(m-1) < x \le 48*m
\end{aligned}
$$

$N_{seg}(x)$ is the number of TCP segments generated by an application message of size x, whereas $N_{cells}(x)$ is the number of ATM cells generated by a CS-PDU of size x. Then the average number of cells generated by an application message can be computed as:

$$\overline{N_{cells}} = \int_0^{\infty} [N_{cells}(MSS + 48)(N_{seg}(x) - 1) + N_{cells}(x - MSS\,(N_{seg}(x) - 1) + 48)]\, f_L(x)dx \quad (2)$$

where $f_L(x)$ is the probability density function of the application message length, which has been assumed exponential with average length L, as already mentioned in 3.2. Finally:

$$\eta_{OH} = \frac{L}{53\,\overline{N_{cells}}} \quad (3)$$

Computing the theoretical goodput as $\gamma_t = \eta_{OH}\, B_{av}$, we can see in fig. 14, that only for very short segments the FCVC curve slightly diverges from the maximum achievable goodput, because of some occurences of unnecessary retransmissions.

4.2 Responsiveness

TCP is a transport protocol whose main goal is improving the reliability of underlying networks, by preventing congestion and recovering from it, at the cost of reducing the throughput. An ATM network controlled by FCVC is almost perfectly reliable and then the TCP flow control mechanisms are seldom used and the effectiveness is not affected. PRCA is not perfectely responsive, thus TCP intervention is required, and even the effectiveness is reduced, due to the transmission of ATM cells of TCP segments already corrupted by cell losses, which need to be retransmitted anyway, and to synchronized segment retransmissions. In order to give an idea of the limited responsiveness of PRCA to congestion, we report in fig. 16, the cell loss probability as a function of the number of TCP connections, for $b_g = 1.3$ and $MSS = 1024\ bytes$. We can observe values of the cell loss probability as high as 10^{-2}, when the network is heavily congested by many TCP connections. Unfortunately cell losses are also highly correlated, thus many TCP segments are likely to be simultaneously affected by such losses, requiring simultaneous retransmissions and causing throughput oscillations.

4.3 Fairness

In order to evaluate which scheme provides a better fairness, we need to define a measure of fairness or, equivalently, of unfairness. For the i-th station we can define the fraction of the offered traffic taken away by the control mechanisms:

$$u_i = \frac{t_i - \gamma_i}{t_i} \tag{4}$$

A control scheme could be defined perfectly fair if u_i is exactly the same for all sources with the same priority level. Therefore the variance σ_u^2 of the random variable u might be used as an unfairness indicator. This unfairness index is plotted in fig. 17, as a function of the number of TCP sources for both schemes under comparison ($b_g = 1.3$ and $MSS = 1024\ bytes$). Obviously, the index is meaningful only in the congested regions. As evident in the figure, in the interval where the congested regions overlap, FCVC seems to be fairer than PRCA. This result can be explained by the following considerations. FCVC, which is based on separated VC buffers, can implement a scheduling policy among VCs (simple round-robin in our simulations), allowing to comply eventually with different priority levels and especially to achieve fairness among users of the same level. With the shared buffers used by PRCA, any scheduling policy is not straightforward to be implemented. Moreover when congestion occurs, all sources are likely to be notified by congestion indications, and consequently their rates are not increased, but only those sources which are affected by actual losses will have their TCP windows shrunk to one segment. They enter a stop-and-wait phase. If the number of cells generated by a single message is less than NRM, they do not have any chance to increase their rate until the next segment can be transmitted. Hence, they recover from congestion very slowly. This effect can be probably smoothed, by observing the system for longer times, but our results allow us at least to assert that PRCA shows more unfairness on small time scales. The last consideration may not be necessarily important if these time scales are only a small fraction of a TCP connection lifetime.

5 Conclusion and open issues

In this paper we have presented an unbiased comparison of the two most debated congestion control schemes for ABR services in ATM LANs running TCP/IP. FCVC features outstanding performances, although it requires a complex and costly switch architecture, which makes it hardly feasible. However, it sets an upper limit of effectiveness, which PRCA should try to approach, without requiring many additional functionalities. First, an appropriate choice of the parameters MDF and AIR may allow to achieve a trade-off between effectiveness and responsiveness. The criteria to be used for this choice should be simple enough to be implemented run-time. Second, a smart way of discarding cells when congestion occurs, can avoid synchronized retransmissions and limit the throughput instability [RF94].

The same comparison should also be performed with larger networks and more complex topologies, where FCVC effectiveness can be limited, because it would be more affected by useless and wasteful TCP retransmissions. Moreover, the effect of including the transmission time in the TCP timers and round-trip samples should be carefully addressed, especially for FCVC which pressurizes ATM cells at the source sites, increasing the transmission times rather then the actual round-trip times.

All these issues are likely to be subject of further studies.

References

[BL90] R. Bagrodia and W. Liao. Maisie: A language and optimizing environment for distributed simulation. In *1990 Simulation Multiconference: Distributed Simulation*, San Diego (CA), January 1990.

[Com91] D. E. Comer. *Internetworking with TCP/IP*, volume vol. I. Prentice Hall, 1991.

[ea94] M. Hluchy et al. Closed loop rate-based traffic management. Technical report, ATM Forum, September 1994.

[GTG93] Mario Gerla, Tsung-Yuan Charles Tai, and Giorgio Gallassi. Internetting LAN's and MAN's to B-ISDN's for connectionless traffic support. *IEEE Journal on Selected Areas in Communications*, 11(8):1145–1159, October 1993.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Proc. of SIGCOMM '88*, Stanford, CA, August 1988. ACM.

[KC94] H. T. Kung and Alan Chapman. The FCVC (Flow-Controlled Virtual Channels) proposal for ATM networks. Technical report, January 1994. ATM Forum report.

[KC95] H. T. Kung and K. Chang. Receiver-oriented adaptive buffer allocation in credit-based flow control for atm networks. To appear in INFOCOM '95, April 1995.

[New93] P. Newman. Backward explicit congestion notification for atm local area netwroks. In *GLOBECOM '93*, Houston (TX), December 1993.

[PF94] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. In *ACM SIGCOMM '94*, August 1994.

[RF94] Allyn Romanow and Sally Floyd. Dynamic of TCP traffic over ATM networks. In *ACM SIGCOMM '94*, August 1994.

[SNS+89] H. Suzuki, H. Nagano, T. Suzuki, T. Takeuchi, and S. Iwasaki. Output-buffer switch architecture for asynchronous transfer mode. In *IEEE International Conference on Communications*, Boston (USA), June 1989.

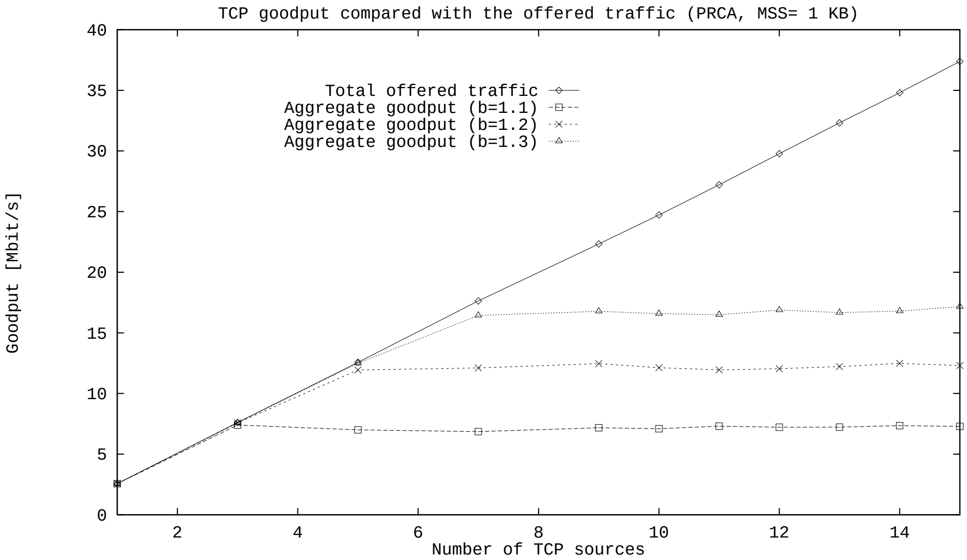

Figure 6: Total TCP goodput with variable b_g (PRCA).

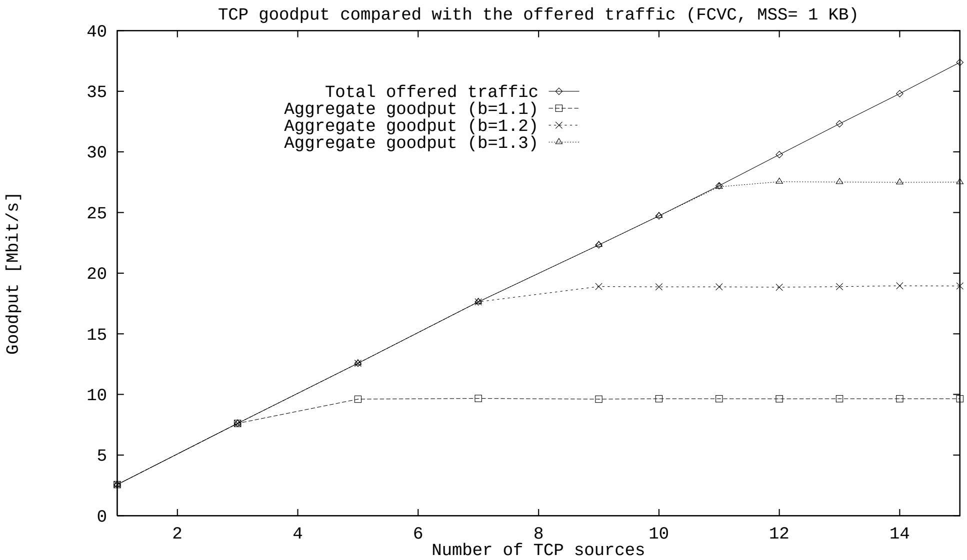

Figure 7: Total TCP goodput with variable b_g (FCVC).

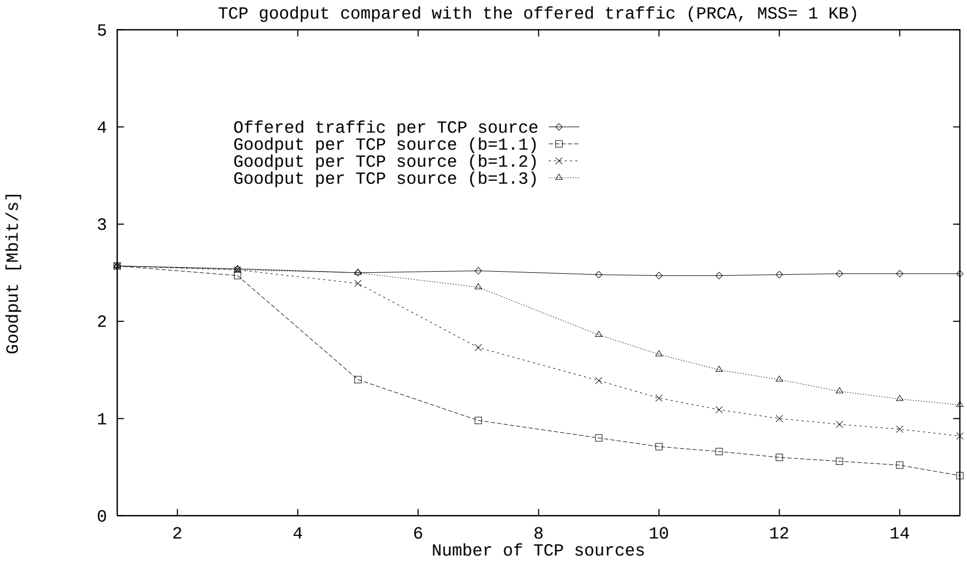

Figure 8: Goodput of a single TCP source with variable b_g (PRCA).

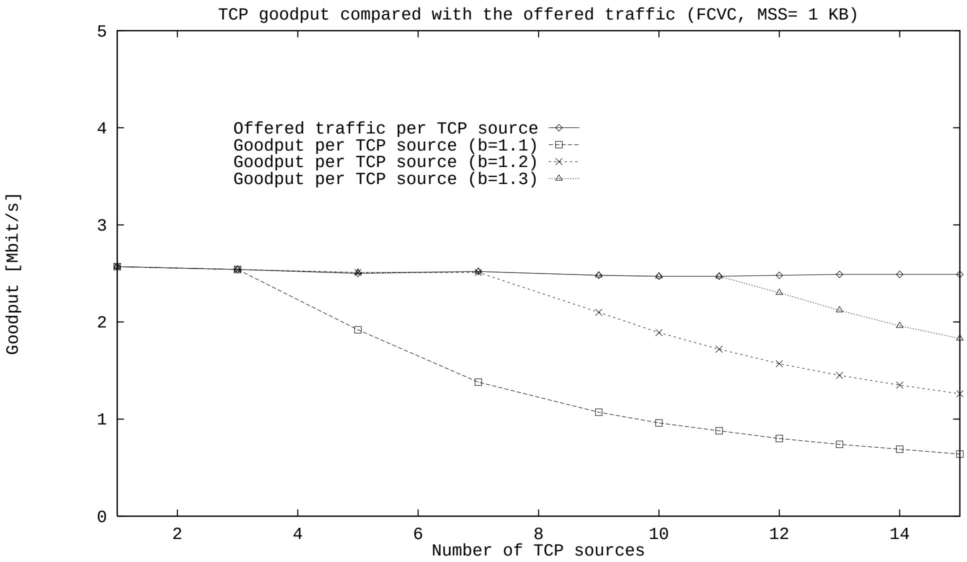

Figure 9: Goodput of a single TCP source with variable b_g (FCVC).

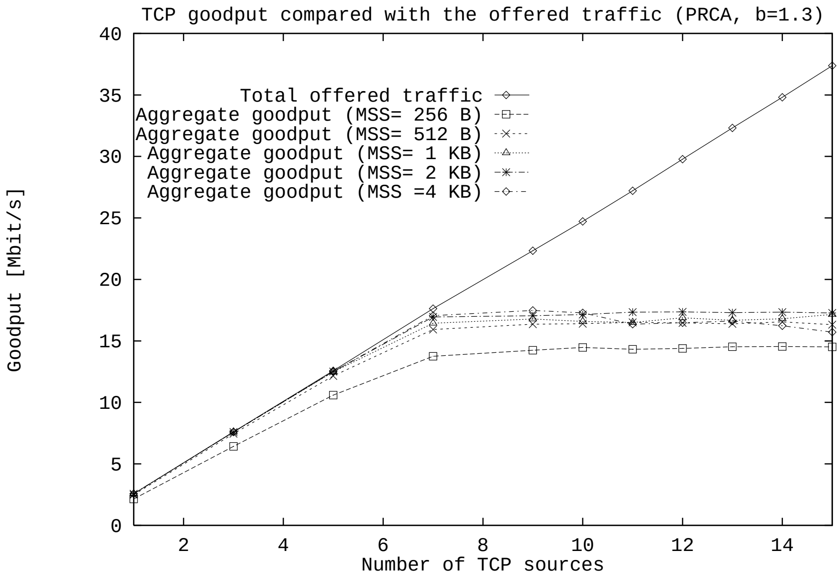

Figure 10: Total TCP goodput source with variable MSS (PRCA).

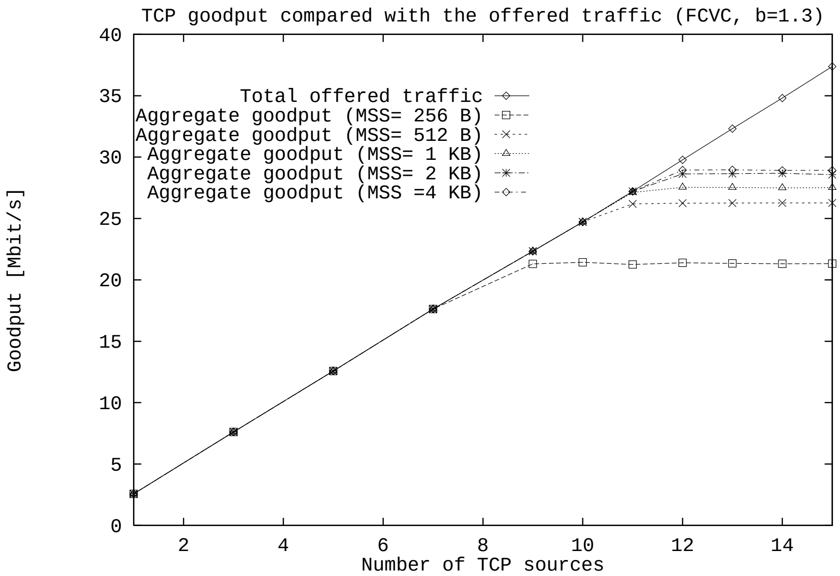

Figure 11: Total TCP goodput source with variable MSS (FCVC).

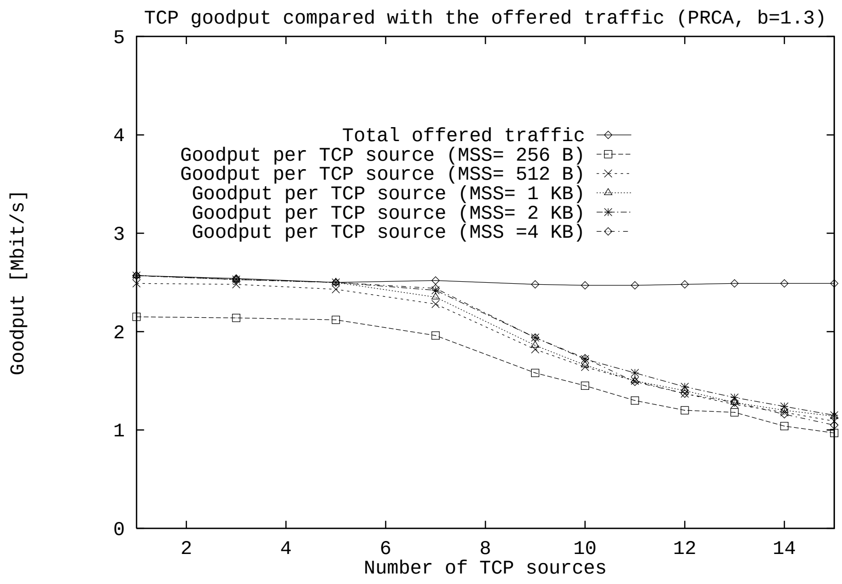

Figure 12: Goodput of a single TCP source with variable MSS (PRCA).

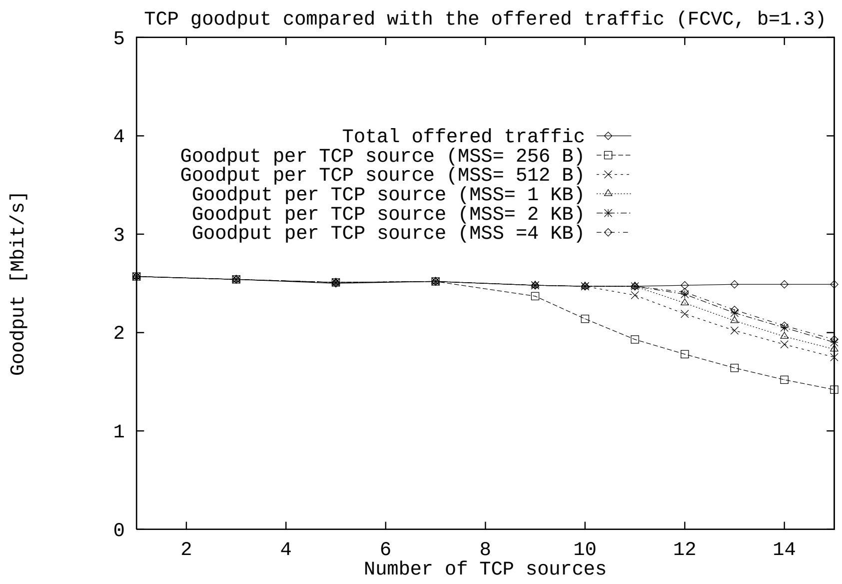

Figure 13: Goodput of a single TCP source with variable MSS (FCVC).

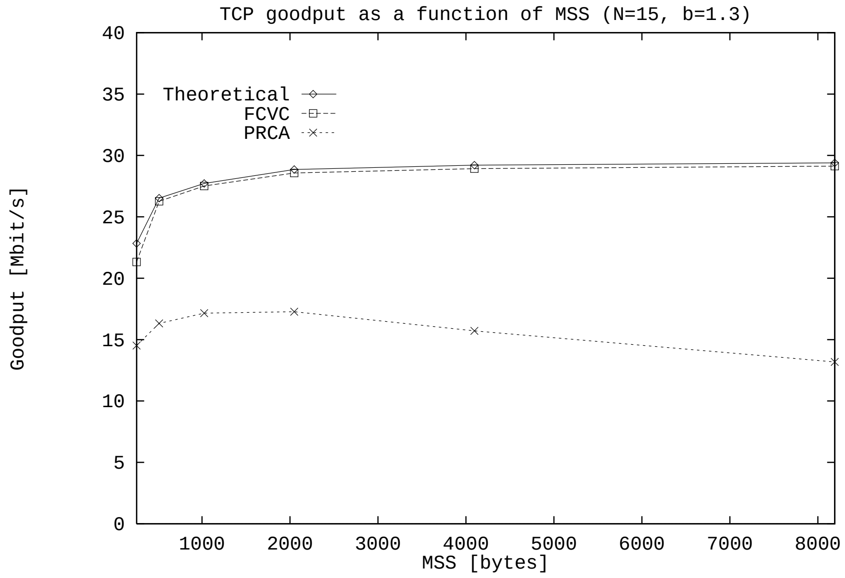

Figure 14: TCP goodput as a function of MSS.

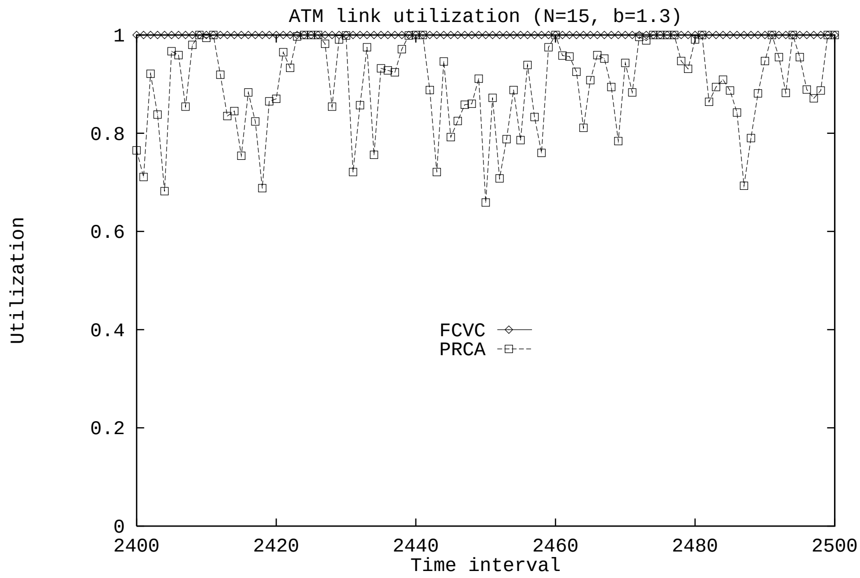

Figure 15: ATM link utilization trace

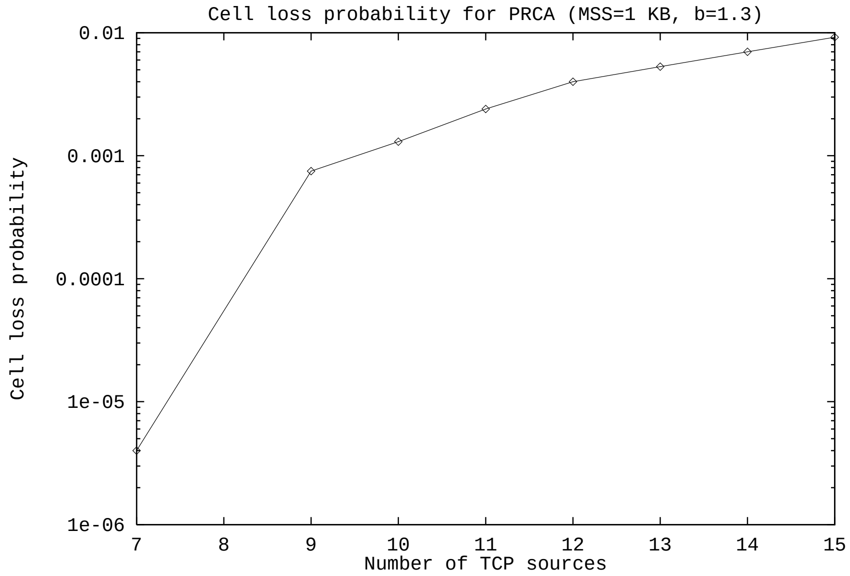

Figure 16: Cell loss probability as a function fo the number of TCP sources.

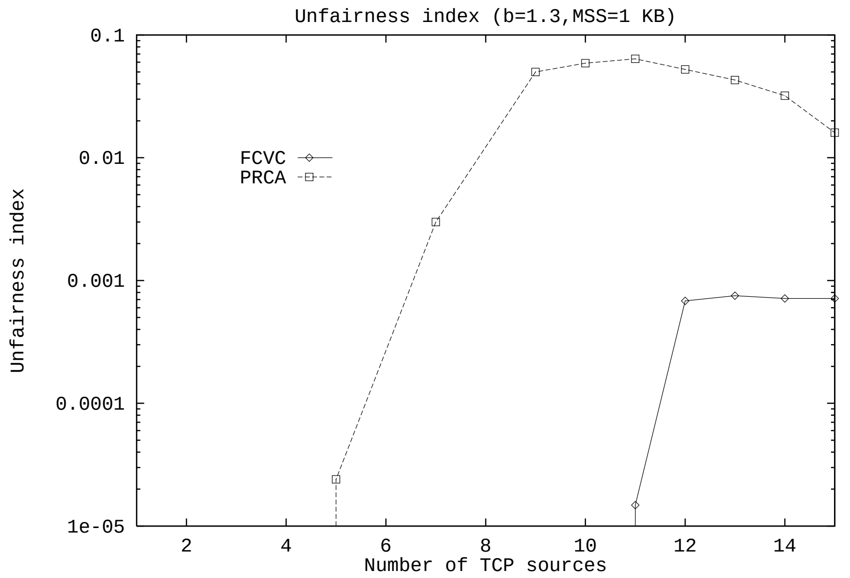

Figure 17: Unfairness index as a function of the number of TCP sources.