ANALYSIS AND DIMENSIONING OF CREDIT-BASED FLOW CONTROL FOR THE ABR SERVICE IN ATM NETWORKS

Rainer Schoenen, Guido Post, Andrea Müller

Institute for Integrated Signal Processing Systems,
Aachen University of Technology (RWTH),
Sommerfeldstr. 24, D-52074 Aachen, Germany
schoenen@ert.rwth-aachen.de, http://www.iss.rwth-aachen.de

Abstract

ATM networks have shown to reveal difficulties in simultaneously guaranteeing the requested Quality-of-Service while efficiently utilizing all the available bandwidth for economic reasons. The ABR (available bit rate) service class was designed to fill the gap of bandwidth with less delay but loss sensitive traffic. Due to the bursty nature of higher-priority traffic and the resulting bandwidth fluctuations a flow control is required for ABR that avoids buffer overflow and cell loss. These demands can both only be supported by credit-based flow control. It offers immediate access to the full link bandwidth without a ramp-up time. However, though its behaviour is deterministic an analysis by queueing models is impossible. In this paper we treat stability, performance and dimensioning issues of a representative class of credit-based flow control protocols with a new stochastic Petri-Net model. We use its formalism to prove its stability and study the utilization performance in all operation regions with it. This implies the proposed dimensioning of buffer sizes and parameters, which is an important task during connection admission.

1 Introduction

ATM is the most promising technology for the next decades of networking. Its success is based on a fast packet-switched network technology supporting different service classes with specific Quality of Service (QoS) parameters and bandwidth demands that are guaranteed by the network. Constant and variable bit rate services support traffic of applications with precisely defined requirements for throughput and delay. The cell loss rate can however often be tolerated up to a certain amount. Thus statistical multiplexing [5] of bursty traffic is desired with a limited probability of cell loss due to buffer overflow. Nevertheless there always remains a considerable amount of spare bandwidth to ensure the low CLR and bounded delay [15].

The ABR service class has QoS requirements that are complementary to those of CBR and VBR: There is no bandwidth and no delay guarantee, but cell loss is critical because of the kind of data that is transported by ABR. Its traffic is assumed to consist of data generated e.g. by FTP, HTTP, NFS or other computer communication protocols that are currently based on TCP/IP in the Internet. If this traffic fills up the spare bandwidth in an uncontrolled way cell loss occurs at a switch when at the same time an output link is busy, the buffers are full and new cells arrive. Higher layer protocols react to cell loss with retransmissions leading to the congestion collapse phenomenon [11] which must be avoided totally. This performance issue of cell loss for ABR is very important for customers, while link utilization is a major concern for providers.

The goal of flow control on the ATM layer is to avoid buffer overflow by providing a closed loop feedback to sending nodes, instructing them to regulate their cell flow according to the semantics of the protocol. A flow control can only be effective and useful if each link is such a closed control loop, because its control delay is only the round trip time of the link and the receiver resources whose overflow is to be avoided are only affected by the traffic on this link.

The rate based flow control mechanisms are unable to jointly utilize all the available bandwidth and avoid all cell loss [16,10]. This work is based on a hop-by-hop credit based flow control protocol which offers the demanded features. Its stability, performance and dimensioning are subject of this paper. We utilize the advantages of the Petri-Net paradigm [17,18] to combine deterministic protocol behaviour in a stochastic environment.

In times where memory is cheap but bandwidth is always scarce individual per-VC buffer is worth its investment. We calculate the required resources necessary for achieving full link utilization.

The paper is organized as follows: Following related work we analyze the credit-based flow control protocol with the Petri-Net paradigm in section 3 and present its performance results in section 4. In section 5 we show exemplary cases for operating regions. Building upon the results of section 4 the dimensioning for admission control of ABR is presented in section 6.

2 Related Work

Existing approaches to ATM flow control [1,3] for the ABR service can be divided into those utilizing rate adjustments at the (virtual) source node initiated by resource management cells from the destination and those keeping a virtual credit for each connection in each switch which must not exhaust during operation.

The majority of published work on rate-based flow control (RBFC) deals different variants of algorithms to instruct the source to increase or decrease its cell rate for a specific connection [1,9,10]. Credit-based flow control (CBFC) algorithms do not need such a variety because its principle is always the same. Buffer overflow is impossible by construction. This property is proven formally in this paper. Variations mainly exist in the way the credit number is calculated and transmitted [4,8,2,6], and methods to cope with lost credit and data cells due to link failures. The basic parameters influencing utilization and control bandwidth are studied in this paper using the Petri-Net formalism, which has not been used in the flow control literature so far.

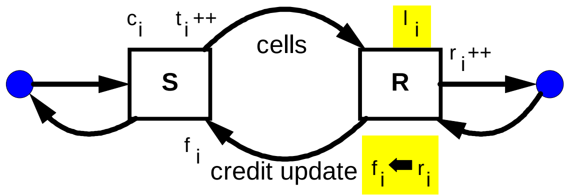

Figure 1: The basic credit flow control loop

3 Credit-based flow control

We assume a CBFC realization very similar to QFC [6], N23 [2] and CUP (FCVC) [4]. The basic operation builds upon on the control of each link in a closed loop between adjacent switches in the network (fig. 1). The buffer in the receiving switch (R) of the size l_{link} is available for temporary storage of cells from the sender (S) during moments of congestion on the output links of (R). It is logically partitioned into individual memory l_i (or $limit_i$) for each connection i out of C ABR connections. (S) knows about the limit l_i of each connection i and installs this as the initial credit. Cells may now be sent arbitrarily (behind higher priority cells) as long as the current credit c_i is positive or zero. For each sent cell a counter of transmitted cells t_i is incremented from zero initially. (R) increments a counter r_i of cells that are received and forwarded to the next node. In regular intervals this counter is sent back to the source (S) which updates its state information for all received connections i by replacing the old values of surely forwarded cells f_i with the new contents $f_i^{(new)}$ of each record. The current credit for connection i is now

$$credit_i = c_i = l_i - t_i + f_i^{(new)} \geq 0 \qquad (1)$$

It is clear that the value r_i in (R) becomes $f_i^{(new)}$ in (S) after at least a time t_p. Also the counter t_i in (S) becomes r_i in (R) after least after a time t_p.

The link length can be defined by the round trip time $t_{rtt} = 2 \cdot t_p$ (propagation time of cells) or equivalently the number of cells currently on the link $c_{rtt}=t_{rtt} \cdot linkrate$.

Because of the deterministic behaviour of the protocol we use a Petri-Net model of the control loop as shown in fig. 2. Here the control loop is modelled for two connections to explain the principles. The cell scheduler [12, 13] which controls the firing sequence of the transitions 'server' is omitted here. According to the Petri-Net paradigm [17] *places* (circles) model a storage location for tokens. The *marking* of a place means the state value of the current number of stored tokens in it. *Transitions* (bars) model actions; when they *fire*, a specified number of tokens is removed from all its input places and all output places get additional tokens. The transitions' execution time distribution is distinguished by their pattern [18]. Black bars mean immediate execution.

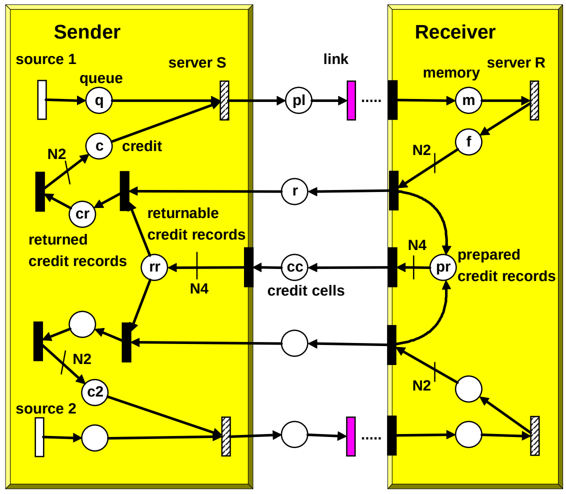

Figure 2: Petri-Net model of the flow control

Initially all but the credit places **c** and buffered ABR cells **q** are empty. Flow controlled ABR cells of a connection i that are queued in place **q** in the sender (S) can now be transmitted (server S) until the credit of c_i tokens in Place **c** is exhausted, exactly as given in eq. (1). After being served a cell token stays on the link for a time t_p until received and put into the memory **m**. After being forwarded to the next switch the cell is counted in **f** (in fact the counter f_i is the number of tokens that ever flew through place **f**). When *N2* tokens of this connection are complete in **f**, a credit record is prepared and stored in **pr**. It takes another quantum *N4* of these records and those of other connections to trigger the generation of a credit cell (which means that *N4* records can be packed into one special ATM RM cell). A credit cell stays on the link for the time t_p before the sender receives it and splits it into separate records in **rr**. The real record must contain the number $f_i^{(new)}$ and the connection identifier i, but this can only be modelled with a coloured Petri-Net. To model this correctly with anonymous tokens an auxiliary place **r** is introduced for each connection. The identified records flow through **cr**, where they are expanded into *N2* new credit tokens for place **c**. The main advantage of this model is its suitability to structural analysis [17] for proving liveness and safe behaviour, as show now.

The purpose of keeping N2 and N4 greater than one is to reduce the frequency of necessary update cells resulting in a lower control overhead. By adjusting these parameters $N2$ and $N4$ the required bandwidth for credit cells on the feedback channel can be adjusted. The ratio of the token rate in forward direction (ABR cell rate) to the token rate backwards (credit cell rate) is exactly:

$$\bar{r}_{control} = \bar{r}_{ABR} \cdot \frac{1}{N2 \cdot N4} \qquad (2)$$

A control overhead share of 5% of the ABR bandwidth in forward direction is established e.g. by $N2{\cdot}N4{=}20$.
The critical loop (for one connection) contains both link directions (places **pl** and **cc**), the memory **m**, the modulo-$N2$ counter **f**, a storage for prepared credit records (**pr**) and the credit buffer **c** (and places **rr** and **cr** that do not contribute). This loop is consistent [17] if we take only one connection into account, which implies that its weighted sum of tokens is constant:

$$pl+m+f+N2{\cdot}pr+(N2{\cdot}N4){\cdot}cc+c =const \qquad (3)$$

The other loop is always consistent and yields:

$$pl+m+f+N2{\cdot}r+c =: limit_i=l_i \qquad (4)$$

This proves that the algorithm is safe, i.e. does not lose cells (neither credit nor data), because its reachability set is constant. And it proves that it is bounded, i.e. the buffer memory usage of this connection is limited by

$$m = l_i - (pl+f+N2{\cdot}r+c) \leq l_i \qquad (5)$$

which avoids buffer overflow and cell loss.
For the protocol to work properly a lower bound of the limit l_i (= buffer size) must hold, because a token flow must be possible in any case, which is called liveness.
The net is conditionally live if for each connection the buffer is at least $l_i \geq N2$. This follows from eq. (4) if an $r{>}0$ should be possible. Conditionally means that other connections must contribute to the completion of a credit cell containing $N4$ records (fill place **pr**). The net is unconditionally live (for each VC i) iff[1] $l_i \geq N2{\cdot}N4$, in which case each critical loop itself is live (eq. (3) permits $cc{>}0$ and $pr{>}0$).
Bounded memory and liveness imply stability of this credit-based flow control protocol. It can also be made robust against lost credit cells by using absolute counters [6] that are valid no matter if the previous credit cell arrived.
In fig. (3) the flow control is extended to three nodes. We observe that each link is such a closed control loop and all loops work independently, because they are only coupled by one transition per connection positioned on both upstream and downstream control loops. What follows is network wide stability.

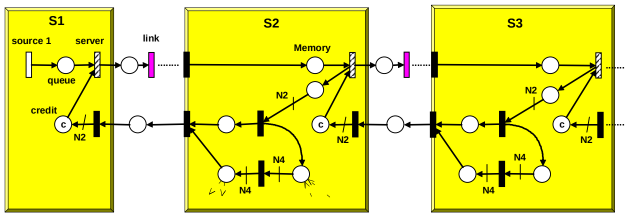

Figure 3: coupled flow controlled links

4 ABR Performance Analysis

The performance of an ABR cell stream is composed by the QoS aspects *cell loss* and *cell delay* as well as the link *utilization* a network provider is interested in. Eq. (5) proves that cell loss due to buffer overflow is impossible. Cell delay, although of less interest for ABR, is always present and compounded by the following terms which constitute a constant and a variable part

$$d_{cell} = d_{propagation}+d_{transmission}+d_{processing}+ d_{switching}+d_{queueing}+d_{flowcontrol} \qquad (6)$$

Queueing delays, i.e. the waiting time for a resource to become available, are usual in distributed transmission systems. The optimum FC performance, however, is reached if no cells are queued *unnecessarily* (in which case the transmission is forbidden but the link is free). This means $d_{flowcontrol}{=}0$, i.e. no additional delay is introduced. This is clearly related to link utilization. Optimum FC performance is equivalent to a fully utilized link independent of the shape of higher priority traffic. At this point all rate-based approaches suffer from rate limitations in a ramp-up phase and fail to instantaneously use free slots on the link [16]. With CBFC the sender can utilize any rate, provided there are enough credits. One requirement for credits to be available is that there is no downstream congestion, i.e. there is no bottleneck ahead in which case backpressure is even desired. The other is that the control time of the flow control loop must be less than the time it takes to transmit as many cells as there is initial credit with a fair share rate. Equivalently the buffer sizes l_i of all connections (which is the initial credit) must be high enough to permit an uninterrupted transmission with a total utilization of 100% even in the absence of higher priority traffic during the control time of the flow control. This control time can be determined with the Petri-Net. The time it takes to return a credit token t_C depends on the round-trip time $2t_p$, the waiting time t_f to complete a credit record (place **f**) and t_{pr} to complete a credit cell (place **pr**).
We are now focusing on the calculation of the maximum possible link utilization u as a function of the initial credit l_i, the number of active connections C and the flow control parameters. Fig. 4 shows the principal dependency. Above a saturation point SP, u=100% is possible. Below SP certain effects have to be taken into account:

[1] If and only if

A) Quantization effect of $N2$: Credit updates arrive in multiples of $N2$, i.e. a new credit record enables the emission of $N2$ further data cells in a quantum burst. The maximum number of cells of a connection transmitted during a cycle of length t_C is

$$b_i := \lfloor l_i/N2 \rfloor \cdot N2, \tag{7}$$

the remainder (l_i mod $N2$) is waiting in place **f** in stationary operation. This leads to

$$t_{f,i} = [N2\text{-}1\text{-}(l_i \bmod N2)]/r_{forward,i} \tag{8}$$

and

$$r_{N2,i} = r_{forward,i}/N2 \tag{9}$$

with $r_{N2,i}$ being the token rate through the transition behind place **f**.

B) Quantization effect of $N4$: This couples independent connections such that all of them contribute to an update message. Therefore

$$t_{pr} = (N4-1) / \sum_i r_{N2,i} \tag{10}$$

With eqn. (8),(10) and $r_{forward,i}$ (equals service rate of transition *server R* for each connection i, controlled by the scheduler) being

$$\sum_i r_{forward,i} = r_{forward,i} \cdot C = r_{f,ABR} \tag{11}$$

because of an equal bandwidth share for all i, we get

$$\begin{aligned} t_C &= 2t_p + t_f + t_{pr} = \\ &2t_p + \frac{N2-1-(l_i \bmod N2)}{r_{forward,i}} + \frac{(N4-1)\cdot N2}{C \cdot r_{forward,i}} = \\ &2t_p + \frac{[N2-1-(l_i \bmod N2)]\cdot C + (N4-1)\cdot N2}{r_{f,ABR}} \end{aligned} \tag{12}$$

This results in the maximum link utilization

$$u_{\max} = \frac{C \cdot b_i}{t_C \cdot r_{link}} = \frac{C \cdot \lfloor l_i / N2 \rfloor \cdot N2}{2t_p \cdot r_{link} + C \cdot [N2-1-(l_i \bmod N2)] + (N4-1)\cdot N2} \tag{13}$$

assuming $r_{f,ABR}$=r_{link}, i.e. a free output link, and same buffer sizes l_i for all of the C connections. It is valid until u_{max}=1 is reached. This result is shown in fig. (4) for several numbers of active connections and parameters $2t_p$=2000/r_{link}; $N2$=21; $N4$=1.

Simulation results using the event-driven simulator OPNET [7] deliver exactly the same graphs.

The ABR sources are modelled as greedy here, i.e. their cell rate demand equals the link capacity. Without this the utilization would be limited by the sources, not by the flow control. This assumption represents also the worst case ABR traffic demand.

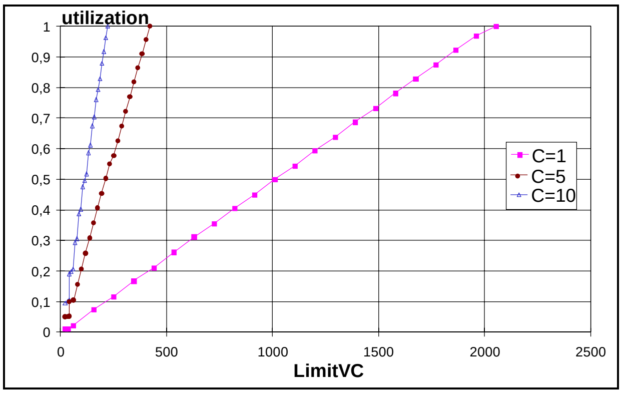

Figure 4: link utilization as a function of credit l_i

Eq. (13) allows both to use l_i to control the maximum output rate and to dimension it such that 100% utilization can be achieved.

Due to the quantization effect of $N2$ we observe a nonlinearity that becomes stronger the smaller l_i and the higher C is. In fig. (5) we see the utilization increase in stairs at l_i being an integer multiple of $N2$. This suggests that l_i should always be chosen such that there is no modulus. With this constraint, eq. (13) reduces to

$$u_{\max} = \frac{C \cdot l_i}{2t_p \cdot r_{link} + C \cdot [N2-1] + (N4-1)\cdot N2}, \tag{14}$$

which can be used to dimension for maximum u_{max}:

$$l_{i,favor} \geq \left\lceil [N2-1] + \frac{2t_p \cdot r_{link}}{C} + \frac{(N4-1)\cdot N2}{C} \right\rceil \tag{15}$$

Eq. (15) also guarantees the liveness condition (eq. 4).

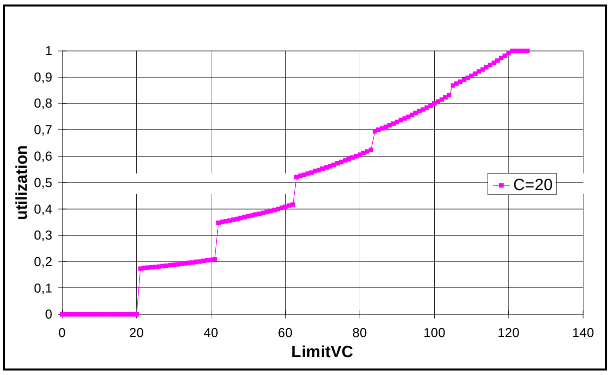

Figure 5: quantization effect of credit feedback

5 Operation Regions and Traffic Patterns

Depending on the dimensioning of l_i some operating regions can be distinguished. Having a free output link (r_{ABR}=r_{link}) and with $l_i > l_{i,favor}$ a full utilization is possible i.e. there is a saturation of utilization. The actual credit never drops to zero because updates arrive just in time to increase it (in quantums of $N2$=21).

With $l_i < l_{i,favor}$ the potential output rate r_{ABR} of switch (R) is higher than the maximum rate allowed by the flow control (eq. 13). The cell flow stops as the credit is exhausted, but after some time new credits arrive and a new burst begins. Each credit update increases the current credit by $N2$, thus the utilization pattern repeats periodically with the burst width b_i given by eq. (7) and a period length t_C (eq. 12). This is the typical traffic pattern observed when we are operating in the increasing region of fig. (4), the „low credit region".

When we now assume the output link being partially congested (e.g. r_{ABR}=0.5·r_{link}), the maximum ABR utilization is also 50% (output constrained). With a sufficient l_i the saturation to 50% can be achieved. In this case the buffer **m** in (R) is usually full. But if we shift the constraint to the flow control by reducing the credit l_i we can control the buffer usage to decrease (l_i=65 in fig. 6). The credit updates now come every 2·$N2$ cell slots, but lead to the immediate emission of $N2$ cells, i.e. the wide burst of b_i cells (eq. 7) is subdivided into smaller bursts of $N2$ cells. But we can also see that the buffer empties itself in spite of the congestion in this situation („low credit region").

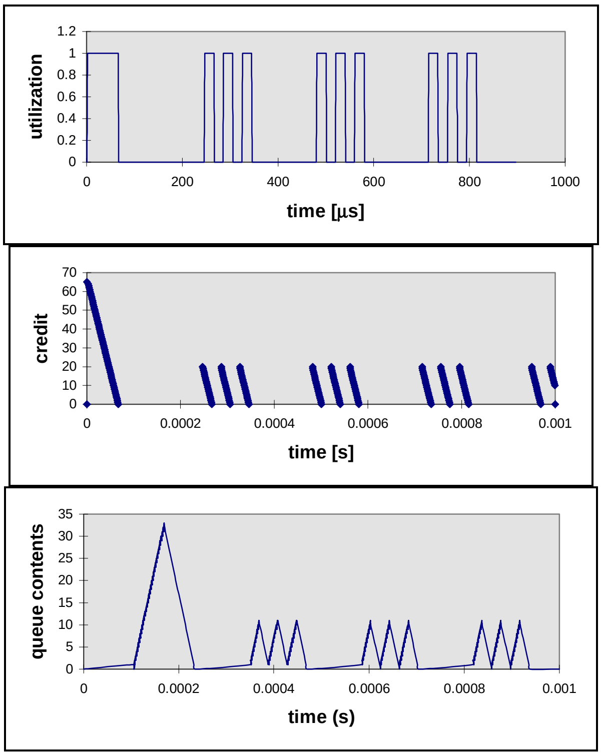

Figure 6: burst behavior with credit depletion

The maximum buffer usage for this case is obtained by considering a fluid-flow model to be

$$b_{max} = (r_i - r_o) \cdot l_i / r_i = (1 - r_o / r_i) \cdot l_i < l_i \qquad (16)$$

which is 32 in fig. 6. After that and before new credits and the next corresponding cells arrive, the buffer contents decrease to zero. This means the buffer is empty before new cells arrive in a burst. This effect may be utilized in an adaptive control algorithm which is not discussed further here.

6 Dimensioning of resources

CBFC utilizes per-VC queueing for maintaining fairness [14]. As eq. (15) shows, the buffer allocation l_i for a connection depends mainly on the link length connected to a switch input. The total buffer needed clearly also depends on the number C of active connections. While eq. (11) assumed an equal share of bandwidth for each connection, it is often more useful to relate the individual buffer l_i to the maximum supportable rate (traffic parameter) PCR_i of the connection. The supported rate of an initial credit l_i is

$$r_{i,\max} = \frac{b_i}{t_C} = \frac{\lfloor l_i / N2 \rfloor \cdot N2}{2t_p + \frac{N2 - 1 - (l_i \bmod N2)}{r_{f,i}} + \frac{(N4-1)\cdot N2}{r_{f,ABR}}} \qquad (17)$$

To support $r_{i,max}$=PCR_i and choosing modulus-free l_i this buffer is required:

$$l_{i,PCR} \geq \left\lceil PCR_i \cdot \left[2t_p + \frac{(N2-1)}{PCR_i} + \frac{(N4-1)\cdot N2}{r_{f,ABR}} \right] \right\rceil \qquad (18)$$

In the worst case $r_{f,ABR}$ can only be as high as PCR_i.

We propose the allocation rule l_i = $l_{i,\ f\cdot PCR}$. This establishes a controlled maximum connection rate of f times PCR. The overallocation (factor f) is useful in a situation where a previously congested link is available again for ABR. The buffers in switch (S) are at most filled with l_i cells, but at worst cells keep flowing in at PCR_i. The time to empty the buffer again is

$$t_{empty} = l_i / ((f-1) \cdot PCR_i) \qquad (19)$$

and if we choose f=2 the transient times of increase and decrease for the buffer are the same.

If we would spend individual buffer l_i such that each connection alone may reach full utilization (eq. 15 with C=1) a total buffer of size

$$\begin{aligned} l_{total} &= C \cdot \left\lceil 2t_p \cdot r_{link} + [N2-1] + (N4-1)\cdot N2 \right\rceil \\ &= C \cdot \left\lceil 2t_p \cdot r_{link} + N4 \cdot N2 - 1 \right\rceil \end{aligned} \qquad (20)$$

cells would be needed.

The PCR related allocation totally only needs

$$l_{total} = C \cdot \left(f \cdot PCR_{mean} \cdot 2t_p + (N4-1)\cdot N2 + N2 - 1 \right) \qquad (21)$$

cells buffer space. But can we make sure the PCR is used by a connection? Of course we can't, but we must allocate for the maximum rate the switch is expected to transport. For ABR it is possible to accept much more connections than it is possible to satisfy at PCR each. Because no throughput guarantee is given to ABR (despite the MCR) it is proposed for economical reasons to accept as much connections C as can be served by the available buffer memory.

Some numerical examples are given in table 1 for a OC-12 link ($r_{link} \approx 1.5 \cdot 10^6$ cells/s).

C	PCR	l_{link}	l_{total}
1000	10Mb/s	10km	1,2 MB
1000	10Mb/s	100km	2,1 MB
1000	10Mb/s	1000km	10 MB
10000	10Mb/s	10km	12 MB

Table 1: buffer resources for ABR - CBFC

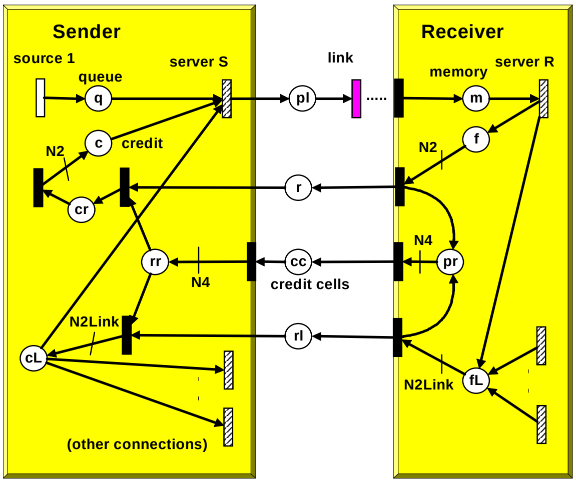

Fig. 7: Buffer sharing by using link flow control

In extreme cases the total memory may be shared between individual connections [6], i.e. it is smaller than the sum of the individual credits ($l_{total}<\Sigma l_i$). This may be used to support more connections with a limited memory, let's say 16MB, on a long distance link.
In fig. (7) this mechanism is shown. An additional credit loop for the whole link exists with an initial credit of l_{total} in place **cL**. This must be greater than zero during operation. Each forwarded ABR cell returns a link credit token to place **fL**, where they wait for the completion of $N2_{link}$ tokens to construct an update record. This record is inserted into the credit cell just like the usual updates for a connection.
The amount of memory mismatch is expressed by a sharing degree between zero and 1-1/C.

$$s = 1 - l_{total}/ \Sigma l_i \qquad (22)$$

Any sharing may however lead to peer blocking, i.e. a number of downstream blocked connections C_{bl} may lead to all other connections being blocked without themselves being congested downstream. The explanation with fig. (7) is that a number C_{bl} of places $\mathbf{m_i}$ hold l_i tokens each due to a downstream congestion, but these $C_{bl}\cdot l_i$ tokens (assuming l_i=*const*) are missing in place **fL** and inhibit the propagation of further cells. Peer blocking does not occur if $C_{bl}\cdot l_i < l_{total}$, which means

$$C_{bl} < C_{max} \cdot (1 - s) \Rightarrow \text{no peer blocking.} \qquad (23)$$

7 Conclusion

We have treated performance issues of credit-based flow control by terms of link utilization and cell loss with a functional Petri-Net model. The stability and liveness is proved by structural properties of this Petri-Net. The influence of parameters as well as dimensioning of buffers are treated by analyzing its timing behavior, which is also discussed in some example scenarios. Adapted to the specific effects and needs of flow-controlled ABR a memory-efficient allocation strategy is derived. It shows that in times when bandwidth is scarce but memory is cheap, credit-based flow control can help sqeezing the most out of ATM.

8 References

[1] Raj Jain, „Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey“, Comp. Networks and ISDN Systems, 1995

[2] H.T.Kung, R.Morris, T.Charuhas, D.Lin, „Use of Link-by-Link Flow Control im Maximizing ATM Network Performance: Simulation Results, Proc. IEEE Hot Interconnects Symposium, 1993

[3] K.K.Ramakrishnan, P.Newman, „Integration of Rate and Credit Schemes for ATM Flow Control“, IEEE Network, March 1995

[4] H.T.Kung, R.Morris, „Credit-Based Flow Control for ATM Networks“, IEEE Network, Vol.9, No. 2, 1995

[5] F.Vakil, „A Capacity Allocation Rule for ATM Networks“, IEEE Globecom 1993

[6] The Flow Control Consortium, „Quantum Flow Control“, http://www.qfc.org/

[7] Opnet Simulation Software Manual, http://www.mil3.-com/Modeler/home.html

[8] C.Özveren, R.Simocoe, G.Varghese, „Reliable and Efficient Hop-by-Hop Flow Control“, ACM SIGCOMM 1994

[9] N.Ghani, J.W.Mark, „Dynamic Rate-Based Control Algorithm for ABR Service in ATM Networks“, Globecom 1996, p. 1074

[10] H.Ohsaki, M.Murata, H.Miyahara, „Robustness of Rate-Based Congestion Control Algorithm for ABR Service Class in ATM Networks“, Globecom 1996

[11] S.Kalyanaraman, R.Jain, S.Fahmy, R.Goyal, F.Lu, S.Srinidhi, „Performance of TCP/IP over ABR Service on ATM Networks“, Globecom 1996

[12] Y.Shimojo, „A Fair Queuing Architecture for ATM Switches with Input Buffers“, Globecom 1996, p. 830

[13] T.Yang, „An Optimal Service Scheduling Policy for Packet Networks with Quality of Service Guarantees“, Globecom 1996

[14] S.Archambault, J.Yan, „Performance Analysis of Per-VC Queuing“, Globecom 1996

[15] G.L. Choudhury, D.M. Lucatoni, W. Whitt, „Squeezing the Most Out of ATM“, IEEE Trans. Communication, Vol. 44, No. 2 Feb. 1996

[16] G. Hasegawa et al., „Performance of TCP over ABR Service Class“, Globecom 1996

[17] Murata, „Petri Nets: Properties, Analysis and Applications“, Proc. IEEE, Vol. 77, No. 9, 1989

[18] M.A.Marsan et al., „Modelling with generalized stochastic Petri Nets“, J.Wiley & sons.