

# Flow Control in ATM based Local Area Networks

A. ABDULMALAK, Y. RAYNAUD

*Université Paul Sabatier, IRIT/SIERA, 118 route de narbonne, 31062*

*Toulouse France Tel: (33) 61 55 67 68, Fax: (33) 61 52 14 58,*

*Email: {abdulmalak,raynaud}@irit.fr*

## Abstract

ATM networks are expected to provide the information transport for a rich mixture of services and applications, associated with which will be a broad spectrum of traffic types and transport performance needs [DePrycker, 1993] [Händel, 1991].

A reliable resource control scheme must be implemented in order to guarantee to the users of the communication infrastructure the *Quality of Service* (QoS) they require. One key element in the management of ATM based network is congestion control.

In this paper we propose a flow control algorithm to be implemented as part of the congestion control architecture in Local Area Networks based on ATM (LATM). This algorithm, called *Adaptive Leaky Bucket* (ALB), implements a hop-by-hop control which is applied on virtual channels that transport *Best Effort* traffic. We use simulation models to show that the ALB algorithm performs well, offering a lossless network while guaranteeing a full link utilization and the QoS required by the users.

## Keywords

ATM, congestion control, flow control, QoS, simulation, Adaptive Leaky Bucket

## 1 INTRODUCTION

The control of congestion in ATM based networks is very important issue that must be properly treated to permit these networks to operate efficiently and provide good Quality of Service (QoS) to all network users. ATM networks are expected to provide the information transport for a rich mixture of services and applications, associated with which will be a broad spectrum of traffic types and transport performance needs. These services have different types of QoS requirements

at considerably varying levels adding a new dimension to the problem.

Congestion indicates the failure of the network resources to meet the clients' demands. Congestion is a user oriented measurement [Abdulmalak, 1993a]; a network might be considered as congested from the perspective of one user and not from the perspective of another.

*So, a user X considers a network as congested if the Quality of Service (QoS) offered by this network doesn't match the QoS required by X.*

We believe that this definition is a very interesting one for the following reasons:

- it takes into account that congestion depends on a user point of view
- a distinction is made between congestion and performance indexes
- QoS degradation will indicate a congestion in a network.

The Bandwidth provided by an ATM network should be shared by the users with respect to their QoS requirements. For this, a reliable traffic control must be implemented.

This subject have occupied many articles over the last few years (e.g. [Cooper, 1990] [Eckberg, 1992] [Gallassi, 1990] [Gilbert, 1991] [Habib, 1991] [Hong, 1991] [Jain, 1990] [Lazar, 1991] [Wernik, 1992] [Yazid, 1992]). The various congestion control schemes can be classified according to the level at which they operate: Connection establishment phase, Cell transmission phase. The main techniques are described in the recommendation I.371 [ITU I371, 1994] and various research papers (for example, [Boyer, 1992] [Roberts, 1991] [Thilakam, 1992]).

In the rest of this paper, we will consider congestion control in Local Area Networks based on ATM (LATM) and we will present the ALB algorithm destined to control the flow and offer lossless LATMs.

## 2 ATM BASED LOCAL AREA NETWORK (LATM)

The need to support real time applications (e.g. multimedia applications) has pushed researchers of both the computer networking community and the telecom community to work on designing new infrastructures based on high speed networking. The ATM technology, which is originally intended for Wide Area Networks, is now making its first steps in the world of LANs. This technique selected to be the key component of B-ISDN architecture will provide LANs the networking support needed to cross the 100Mbit/s threshold. ATM based LAN offers a base rate of approximately 155Mbit/s with the possibility to use higher speeds of multiple of the base rate. This will allow the coexistence of both traditional data communications and real time applications.

Moreover, ATM will simplify the interconnectivity of local area networks via the telecommunication systems.

LATM designers must consider two major requirements: Cost and Maintenance. The price of LATM terminals and switches should not be much more expensive than an actual network connection (e.g. an FDDI connection). And the maintenance of ATM-LAN equipments must be undertaken by management applications which can be integrated within the private network management tools using existing standards (e.g. SNMP, CMIP).

It is a big challenge for ATM-LAN deployment to provide a high level of QoS to the customer while maintaining a low-cost and simply-manageable design.

In LATM we distinguish two major QoS classes [Newman, 1994]:

☞ **Guaranteed**

☞ **Best Effort**

The *guaranteed class of service* will support traffics for which an explicit guarantee of QoS is given. And the *Best Effort* class of service is used to support applications like data traffic with no stringent QoS requirements. The *Best Effort* service is sometimes referred to in the literature as *elastic service*.

In order to provide a better service to data traffic, the ATM Forum has classified this traffic into two categories [ATMForum, 1994a] [ATMForum, 1995]:

☞ **Unspecified Bit Rate (UBR)**: This class is intended for traffics with no delay or loss requirements. There is no QoS guarantee. In [ATMForum, 1995], the UBR is considered as an interpretation of the term “*Best Effort service*”.

☞ **Available Bit Rate (ABR)**: This is an improved version of the UBR service. It is defined to be used for applications that are sensitive to loss, but can tolerate variations in delay. The traffic generated by this service will be flow controlled. A Minimum Cell Rate (MCR) is negotiated with the network.

These definitions are not definitive. The UNI 4.0 of the ATM Forum should contain the final version. A great debate has been going on around the definition of the ABR service, and lots of propositions have been made, for example:

- ☞ ABR with MCR = 0
- ☞ Two classes of ABR [ATMForum, 1994f]:
  - ABR+ with MCR >0
  - ABR0 with MCR = 0
- ☞ Let the ABR support “pseudo real time” requirements !!!!!
- ☞ Define ABR as a Constant Bit Rate (CBR) service with lower quality
- ☞ In a contribution to the ATM Forum [ATMForum, 1994c], the LAN Emulation SubWorking Group (LANE SWG) has listed some of the requirements relating to an ABR-like service. In this contribution, the LANE SWG asked for a service supporting a MCR=0 and a Peak Cell Rate (PCR) equal to the capacity of the physical link.
- ☞ ...

In this paper, we consider traffic with MCR = 0 and PCR equal to the link capacity. The class of service supporting this traffic is designated by the term *Best Effort*. This traffic will be subject to a flow control.

Flow Control is one of the several mechanisms used to avoid congestion within communication networks. Its main function is to regulate the rate of data transmission between two communicating devices, protecting by that the resources at the receiver side and inside the network. Flow Control

can be applied on a hop-by-hop or end-to-end basis.

In a LATM communication network cell loss would trigger a retransmission mechanism at the higher layers. It has been shown that retransmission is very harmful to network operation causing a quick performance degradation [Abdulmalak, 1993b] [ATMForum, 1993b]. To guarantee a 0% cell loss, we consider a hop-by-hop connection-based flow control mechanism, which will allow any upstream node to send cells if and only if there is a room in the destination buffer of the downstream node. It is obvious here that the implementation of such a flow control mechanism will require buffer space (per connection) proportional to the link length connecting the two nodes:  $Buffer\text{-}Space \geq Throughput \times Round\ Trip\ Delay$ .

Each node in the network should regulate the traffic flow from the different connections, by forcing them to slow down (or eventually stop) their transmission when the number of free rooms in the buffer reaches a certain threshold. The control procedure launched at a specific node in the network will propagate on the upstream nodes to reach the source. As a matter of fact, the receipt of a control message on a node asking it to slow down (or stop) transmission will cause the buffers at this node to fill up, forcing its upstream nodes to slow down.

In the next section, we present the ALB algorithm that we have defined to control the *Best Effort* traffic and to offer a lossless network.

### 3 THE ADAPTIVE LEAKY BUCKET (ALB) ALGORITHM

The Leaky Bucket (LB) method is a common example of a traffic control at the cell level. This mechanism is one way to ensure that sources do not exceed the negotiated rate. Schematically, it can be represented as a token pool with a capacity  $M$  which can be seen as the maximum allowable burst length (not considering tokens generated during the burst); a cell can enter the network only when it can draw a token from the pool; if the token pool is empty, then the cell is discarded, put in queue, or marked and transmitted. Tokens are generated at a certain admission rate  $r$ . After filling the pool, additional tokens are discarded. LB is a simple algorithm which is easy to implement.

The LB mechanism was first introduced in [Akhtar, 1987] as a scheme to police and regulate the traffic flow submitted to the network. With the LB scheme, two types of enforcement action can be taken over the cells arriving at a time when the bucket is empty: either the cell is dropped, or it is marked and injected in the network [Bala, 1990]. This gives rise to several versions of the LB scheme.

We have tested the LB mechanism with a simulation model described in [Abdulmalak, 1993b]. In the simulation we have used an 8x8 input-buffered switch model corresponding to an experimental Hub for LATM, which has been under construction at the research laboratories of Hewlett-Packard at Bristol.

The results have showed that the LB scheme brings some improvement but it can't guarantee a 0% loss. The traffic model we have used is based on Client/Server scheme; five clients and one server have been used. The traffic coming from the five clients towards the server creates cell contention on the output port attached to the server. This causes a modification of the service time of the cells generated by the clients. And the token regeneration rate (which indicates the admission

rate of the network), initially equal to this service time, won't change. To cope with this, we have introduced some "intelligence" within the LB mechanism, in the sense of enabling it to adapt itself to the changes in the service time of a cell within the switch. The idea consists in an exchange of control cells between the switch and the device implementing the LB mechanism.

This approach consists in representing the LB as a "credit" mechanism, since it exercises a control over the sender restricting the maximum amount of data that might be transmitted at a certain instant of time. This will give rise to the Adaptive Leaky Bucket algorithm.

The ALB uses a counter (representing the number of available tokens) which is initially equal to a Maximum counter value, the so-called bucket size. Each time a cell is sent into the network, the counter value is decremented by one. So the counter value is considered as the available tokens. New tokens are generated by periodically incrementing the counter value by a certain amount (the maximum number of available tokens is equal to the bucket size). The ALB mechanism is then described by the initial number of tokens, an update function and an algorithm for the regeneration of the tokens.

The ALB flow control algorithm is implemented on a hop-by-hop basis. Each virtual channel connection (VCC) is divided into several virtual channel links (VCL). To each VCL are associated a sender and a receiver. The sender will only transmit cells if tokens are available and the receiver will send back, periodically, control cells indicating the available buffer space for this VCL. To facilitate the comprehension of the algorithm, we consider, in the following, two consecutive nodes  $i$  and  $j$  and an opened VC between them. The node  $i$  is the sender and the node  $j$  is the receiver.

Two types of ATM cells are used: *Data\_Cell* and *Token\_Cell*. A *Data\_cell* is used to transport data belonging to the VC. A *Token\_cell* is a control cell used to transport information related to the tokens regeneration mechanism.

Node  $j$  will generate a *Token\_Cell* after the forwarding of  $N_{\text{Forward}}$  cells belonging to the VC. This cell will contain the number of free buffers for this VC on node  $j$ . This value will be used by the node  $i$  to regenerate the tokens.

When the node  $i$  receives a *Token\_Cell* it regenerates the tokens by subtracting a value  $X$ , corresponding to the number of cells sent during the last Round Trip Time (RTT), from the value carried by the *Token\_Cell*. As a matter of fact, the value representing the number of free buffers at node  $j$  will be obsoleted as soon as the *Token\_Cell* reaches node  $i$ , since the propagation delay between the two nodes is not negligible (see Figure 1).

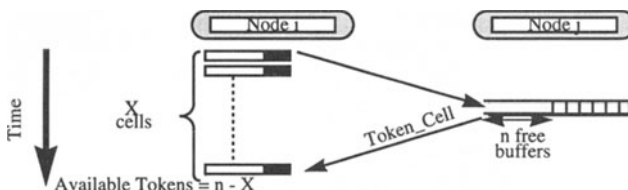


Figure 1 Tokens regeneration.

In order to allow the node  $i$  to sustain its throughput and not to be blocked, the *Token\_cell* should reach it before it runs out of tokens. In other terms, the initial number of tokens should be equal to

$N_{ij} + N_{\text{Forward}}$ , where  $N_{ij}$  represents the RTT interval<sup>1</sup> in cell time. On the other hand, to allow the network to offer a lossless service, the same value of buffers should be reserved on node  $j$ .

### Algorithm

#### a) Sender side

##### for each virtual channel

Number\_Of\_Cells\_Tx :        number of cells sent by the sender  
 Max\_Tokens :                maximum number of tokens  
 Available\_Tokens :         available tokensnombre

##### Initialization

```
Number_Of_Cells_Tx <-- 0;
Available_Tokens <-- Max_Tokens;
Waiting_For_Tokens <-- FALSE;
```

##### Transmission of a Data Cell : "cell"

```
BEGIN
  IF Available_Tokens > 0
    Send(cell);
    Decrement(Available_Tokens);
    Increment(Number_Of_Cells_Tx);
  ELSE
    Waiting_For_Tokens <-- TRUE; #Waiting for tokens
    Sleep(until Available_Tokens > 0); #the sender is blocked
    Waiting_For_Tokens <-- FALSE;
  END IF;
END.
```

##### Receipt of a Token Cell : "cell"

#This cell contains the number of cells the downstream node has forwarded.  
 #This parameter is represented by the variable cell-->Cells\_Fwd

```
BEGIN
  Available_Tokens <-- Max_Tokens + cell-->Cells_Fwd - Number_Of_Cells_Tx;
  IF Waiting_For_Tokens = TRUE
    Wakeup(transmitter);
  END IF;
END
```

#### b) Receiver side

##### for each virtual channel

Number\_Of\_Cells\_Rx :        number of cells received  
 Number\_Of\_Cells\_Fwd :      number of cells forwarded  
 Max\_Tokens :                maximum number of tokens for the VC  
 When\_To\_Send\_Token\_Cell : period used to send token cells

---

1. The RTT interval represents the RTT and the time needed to process the cells on the nodes.

Initialization

```
Number_Of_Cells_Fwd <-- 0;
```

Receipt of a Data Cell: "cell"

```
BEGIN
  Insert(cell) in buffer;
  Increment(Number_Of_Cells_Rx);
END.
```

Forward of a Data Cell: "cell"

```
BEGIN
  Cellule <-- GetFromBuffer(cellule);
  Send(Cellule);
  Increment(Number_Of_Cells_Fwd);
  IF ((Number_Of_Cells_Fwd mod When_To_Send_Token_Cell) = 0)
    SendTokenCell;
  END IF;
END.
```

Sending of a Token Cell: "cell"

```
BEGIN
  (cell->Cells_Fwd) <-- Number_Of_Cells_Fwd;
  SendToUpstream(cellule);
END.
```

## 4 PERFORMANCE EVALUATION AND VALIDATION

In the following, we will study some properties of the ALB algorithm (a more complete study is described in [Abdulmalak, 1995a]). A simulation approach based on the object oriented simulation language MODSIMII, is used to verify some of these properties.

### *Robustness against cell loss and corrupted control cell*

The code presented above shows that the sender can't send cells if there is no space on the receiver buffer, which will guarantee a transmission with no cell loss (we don't consider cells lost due to transmission errors).

A corrupted or lost (due to a transmission error) *Token\_Cell* cell won't have a negative impact on the algorithm operation and that is due to the fact that it uses an absolute update which generates tokens independently from the number of tokens available in the bucket. Deadlock situations can occur when a *Token\_Cell* is lost and the node  $j$  doesn't have  $N_{\text{Forward}}$  cells to forward. To avoid this, we use a control cell called *Ask\_For\_Tokens* which is sent by the sender towards the receiver asking him to indicate the buffers state. For a formal validation of this property, confer to [Abdulmalak, 1995b].

### *Tokens update rate and impact of control cells on data traffic*

The value of the parameter  $N_{\text{Forward}}$  determines the time interval used by a node to send *Token\_Cell* towards an upstream node. The link between the current node and the upstream node will contain a *Token\_Cell* once each  $N_{\text{Forward}}$  *Data\_Cell*. So whenever two VCs, flowing in opposite direction, share a link, data and tokens cells should compete for bandwidth.

For a given value,  $\beta$ , of  $N_{\text{Forward}}$ , the overhead generated by the control traffic is limited to the fraction  $1/\beta$  of the link capacity. A small value of  $N_{\text{Forward}}$  will generate more *Token\_Cells* and thus decrease the bandwidth offered to data traffic, but It will increase the robustness of the ALB algorithm against eventual loss of *Token\_Cells*.

If we consider the case where two VCs, flowing in opposite direction share, the same link, and both VCs represent greedy sources capable of sustaining the full link bandwidth, only *Token\_Cells* can prevent them from sending at full speed. Let  $B_1$  the bandwidth of VC1 and  $\beta$  the value of  $N_{\text{Forward}}$ ; we can write  $B_1 = \beta / (\beta + 1)$ . We have run some simulations using a configuration with two VCs flowing in opposite direction and sharing the same link. The results have confirmed this equation.

### Control cells propagation and HOL Blocking

One of the main problems with *Closed Loop* schemes is the time needed to propagate the congestion indication towards the source [Hong, 1991] [Vakil, 1991]. The use of a hop-by-hop mechanism will reduce the impact of this problem on the network. This simulation studies the propagation of control cells through the network and shows that the ALB algorithm acts fast enough to guarantee a reliable transmission service.

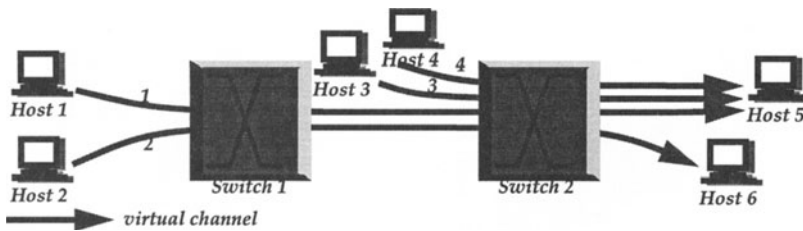


Figure 2 Configuration 1: two switches in series with three VCs towards a single destination

With configuration 1, we have two switches in series connected by a 10km link at a rate of 100Mbit/s. Two hosts, *Host1* and *Host2*, are connected to switch1, and two others, *Host3* et *Host4*, are connected to switch 2. *Host1*, *Host3* et *Host4* communicate with a single host '*Host5*'. *Host2* transmits cells towards *Host6*.

The sources used to generate traffic for *Host1* and *Host2*, are "greedy sources" [ATMForum, 1994a]. They always send the maximum cells permitted and start at the beginning of the simulation. Hosts *Host3* and *Host4* are greedy sources too but they have a delayed turn-on. In this simulation, they start cells generation 2 milliseconds after the beginning of the simulation; they represent sources of type "staggered" [ATMForum, 1994a].

Figure 3 shows the throughput of both hosts, *Host1* and *Host2*. Both traffics are sent on the link connecting the two switches. Thus the bandwidth of this link is divided into two equal parts (the throughput of each host corresponds to 1/2 of the capacity of the link). But after a time  $T$  equal to



2ms, *Host3* and *Host4* start cells generation. At this moment, the three hosts, *Host1*, *Host3* and *Host4*, share the bandwidth of the link connecting *Switch2* to *Host5*. The throughput of each hosts is equal to 1/3 of the capacity of the link.

So the ALB algorithm will allow the network to indicate to *Host1* its new rate, offering thus the occasion to *Host2* to increase its rate and occupy the bandwidth freed by *Host1*.

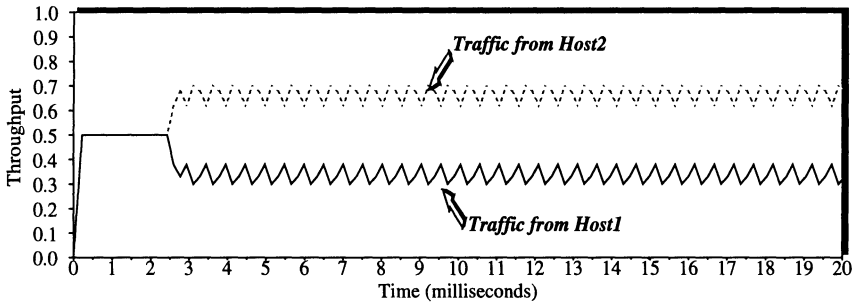


Figure 3 Throughput of both hosts Host1 and Host2, leaving switch 1

These results show the propagation of congestion notification through several switches in order to adapt the sources rates according to the state of the network. The ALB algorithm prevents *HOL Blocking* situation when several virtual channels share the same physical link, and one of those VCs experience congestion on its way. The ALB algorithm guarantees that sources of type 'staggered' get their part of the bandwidth.

### Resource sharing between two virtual channels with two different priorities

The main goal of the ALB algorithm is to allow the network to guarantee the QoS required by the users and to offer a good utilization of the links.

As we have mentioned above, the ALB will be applied to *Best Effort* traffic and will adapt their throughput according to the state of the network. The QoS required by the high priority traffic should not be affected by the cells flows generated by the *Best Effort* service. This simulation shows the efficiency of the ALB algorithm to guarantee the QoS required and to offer to the *Best Effort* traffic the possibility to occupy the unused bandwidth in order to realize a full link utilization.

The network configuration used in this simulation is depicted in figure 4. *Host1* and *Host2* transmit cells towards *Host3* through one switch and using two virtual channels, VC1 and VC2 respectively. The traffic transmitted on VC2 is not submitted to a flow control, and it has a high level of priority. On the other hand, the traffic on VC1 is controlled by an ALB algorithm.

*Host1* represents a greedy source. It transmits cells with a maximum throughput equal to the capacity of the physical link.

*Host2* represents a video source generating high priority traffic which starts cells transmission 1ms after the beginning of the simulation. This traffic is generated using a single trace of an AVA-

200 ATM camera. This trace contains traffic measurements<sup>1</sup> gathered on the Fairisle network at the university of Cambridge UK. The camera was set up to transmit 25 frames per second, JPEG compressed, 24 bits per pixel colour video from normal BBC TV transmission. The spacing policer which limits the peak emission interval of cells from the camera was switched off. Thus the camera transmits cells whenever they are ready to leave the device, and back-to-back cells are possible. The trace comprises both action scenes and relatively static portions.

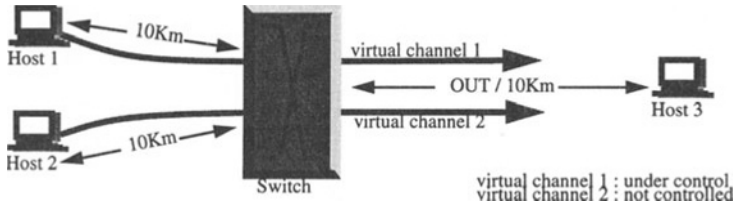


Figure 4 Configuration 2: two virtual channels with two different levels of priority

The switch implements a Round Robin like service discipline with two levels of priority. Two queues are used: one for the traffic with high priority, and one for the traffic with low priority. In case of contention, the switch will always serve the queue with high priority first.

The link utilization is shown in figure 5. We can see that the traffic of VC1 (low priority) occupies the bandwidth left by the VC2 (high priority). There have been no cell loss during the simulation for both VCs. When VC2 asks for more bandwidth, it gets it, and the throughput of VC1 is adapted accordingly. And when VC2 frees some bandwidth, VC1 immediately increases its throughput.

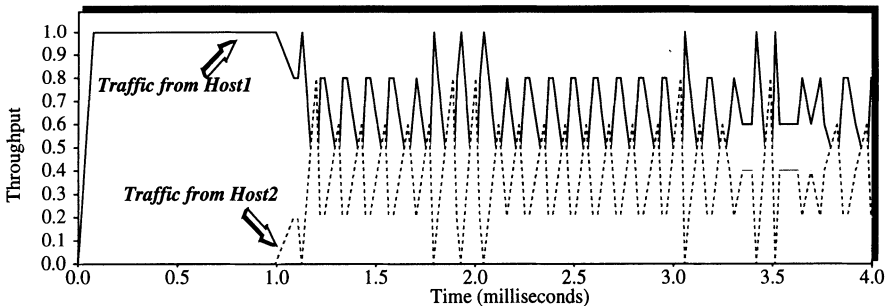


Figure 5 Traffic 2 gets its QoS required, and traffic 1 occupies the unused bandwidth

The ALB algorithm guarantees a full link utilization. This is shown in figure 6. We can note the flexibility of ALB when it comes to bandwidth allocation on the link. These results show the ability of the ALB algorithm to adapt the traffic according to the state of the network.

1. This data is Copyright Simon Crosby and Ian Pratt, 1994, University of Cambridge, UK.

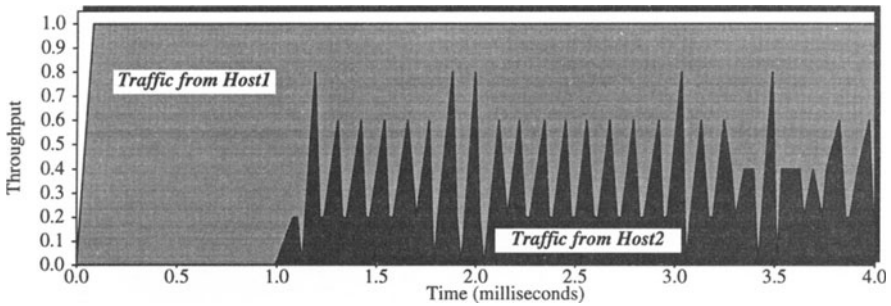


Figure 6 Full link utilization

*Fairness test*

This last simulation is used to test the fairness of bandwidth allocation when using traffic control mechanism (see figure 7). The configuration is based on the *Generic Fairness Configuration (GFC)* defined by the traffic management working group of the ATM Forum.

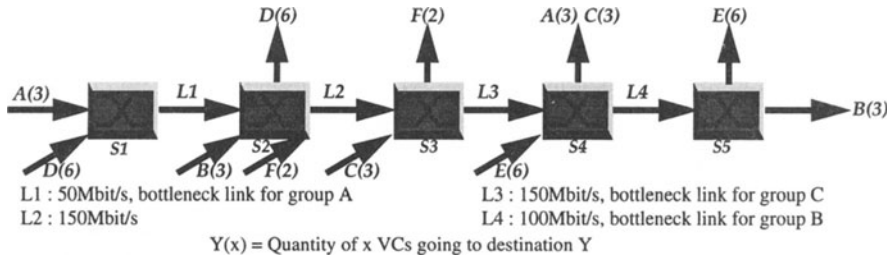


Figure 7 Configuration 3: Generic Fairness Configuration

With this configuration, we have a set of cascaded links experiencing congestion. In particular, there is a congested link that has VCs that are also experiencing bottlenecks upstream and downstream from the link. What is interesting to study with this configuration is the ability to achieve fairness in the steady state.

There are six groups of VCs, “A” through “F”. The A and the B VCs experience three links in congestion. The remaining VCs are employed to insure that each switch and link is experiencing full utilization and congestion. For full congestion, and under the assumption that a VC’s share of bandwidth from the configuration should be that of its minimum share when considering each link individually, the calculated fairness values are given in [ATMForum, 1994a]. The goal of this simulation is to show that the implementation of the ALB algorithm allows the network to offer to the different classes of VCs, the same throughputs calculated in the steady state.

The distance between the switches and the one between the sources and the switches is equal to 1Km.

Figure 8 shows the results for the two groups A and F, obtained from this simulation. The throughputs calculated for these groups are respectively 0.037 and 0.33. We can see that at a steady state the throughputs obtained from the simulation for these two groups correspond with the ones calculated.

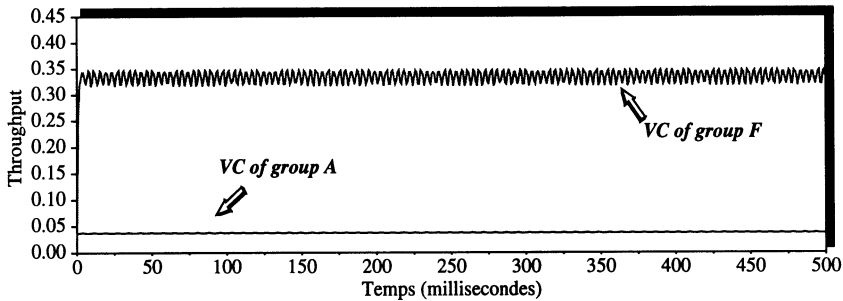


Figure 8 Throughput of both groups A and F

This last configuration allowed us to verify the fairness of the ALB algorithm. For this, we have used the model provided by one of the contributions of the ATM Forum in order to evaluate the fairness index. The simulation's results were very satisfactory. At a steady state, each VC has obtained its calculated optimal throughput.

## 5 Conclusion

In this paper, we have presented the ALB algorithm defined to implement a flow control scheme for *Best Effort* traffic. The main objective of this algorithm is to offer a lossless network and to control the congestion, and thus, to guarantee the QoS required by the users.

The basic idea is that the sender will only transmit cells if and only if the receiver is able to process them. It is a *Closed Loop* credit-based scheme, and it is implemented on a hop-by-hop per VC basis. The simulations we have presented in this paper show the good performance of the ALB algorithm.

Within the ATM Forum, there has been a great debate on the choice of the mechanism to control the flow of ABR traffics. Two main techniques have been discussed:

- ⇒ *Credit-Based* [ATMForum, 1993a] [ATMForum, 1994d] [Kung, 1994] [Ozveren, 1994]
- ⇒ *Rate-Based* [ATMForum, 1994b] [ATMForum, 1994e]

This issue was intensely studied and the result was the adoption in principle of a *Rate-Based* approach to implement ABR flow control. Several ATM Forum contributions have showed that *Credit-Based* mechanisms perform better than *Rate-Based* algorithms, but "Cost" was the main reason for such a decision.

We believe that member companies are not bound by decisions of the ATM Forum, and *Credit-Based* mechanisms will still find place in the market place. The Digital *FLOWmaster Credit-Based* flow control algorithm is a good example.

The control scheme presented in this paper is implemented at the data transmission phase level. In order to build a complete congestion control scheme, this control needs to be augmented with solutions for the connexion admission phase and the long term phase. The work to finalize and build a complete infrastructure for congestion control is in progress, and more results will be reported at a later date.

## 6 References

- [Abdulmalak, 1993a], Abdulmalak, A. and Raynaud, Y. (1993) Quality of Service & Traffic Control in ATM based Local Area Networks. *Proceedings of the 4<sup>th</sup> IFIP/IEEE DSOM93*.
- [Abdulmalak, 1993b], Abdulmalak, A. (1993) Performance and Modelling Study of the Sapphire Hub: Definition of a Traffic Control Mechanism. *Internal Report IRIT-SIERA/Hewlett-Packard, 93/05*.
- [Abdulmalak, 1995a], Abdulmalak, A (1995) Congestion Control in ATM Local Area Networks: Design and Validation of the Algorithm "Adaptive Leaky Bucket". *PhD thesis, University Paul Sabatier Toulouse, May*.
- [Abdulmalak, 1995b], Abdulmalak, A. and Bruel, J.-M. (1995) Performance Evaluation and Validation of a Congestion Control Algorithm for ATM Local Area Networks. *3rd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, July*.
- [Akhtar, 1987] Akhtar, S. (1987) Congestion Control in a Fast Packet Switching Network. *Master's thesis, Washington University*.
- [ATMForum, 1993a] ATM Forum (1993) Digital Flow Control. *ATM Forum / 94-0778*.
- [ATMForum, 1993b] ATM Forum (1993) TCP Over ATM: Some Performance Results. *ATM Forum / 93-0784*.
- [ATMForum, 1994a] ATM Forum (1994) Baseline Text For Traffic Management Sub-Working Group. *ATM Forum / 94-0394R5*.
- [ATMForum, 1994b] ATM Forum (1994) Closed-Loop Rate-Based Traffic Management. *ATM Forum / 94-0438R2*.
- [ATMForum, 1994c] ATM Forum (1994) LAN-Emulation's needs for Traffic Management. *ATM Forum / 94-0533*.
- [ATMForum, 1994d] ATM Forum (1994) Credit-Based FCVC Proposal for ATM Traffic Management Revision 2. *ATM Forum / 94-0632*.
- [ATMForum, 1994e] ATM Forum (1994) Enhanced PRCA (Proportional Rate-Control Algorithm). *ATM Forum / 94-0735R1*.
- [ATMForum, 1994f] ATM Forum (1994) Proposed Modification to Traffic Management Baseline Text. *ATM Forum / 94-1201R1*.
- [ATMForum, 1995] ATM Forum (1995) DRAFT ATM Forum Traffic Management Specification Version 4.0. *ATM Forum / 95-0013*.
- [Bala, 1990] Bala, K., Cidon, I. and Sohraby, K. (1990) Congestion Control for High Speed Packet Switched Networks. *Proceedings of INFOCOM '90*.
- [Boyer, 1992] Boyer, P., Guillemin, F., Serval, M. and Coudreuse, J.P. (1992) Spacing Cells Protects and Enhances Utilisation of ATM Network Links. *IEEE Network, September*.
- [Cooper, 1990] Cooper, C.A., and Park, K. (1990) Toward a Broadband Congestion Control Strategy. *IEEE Network, May*.
- [DePrycker, 1993] M. De Prycker (1993) Asynchronous Transfer Mode: Solution For Broadband ISDN - 2<sup>nd</sup> edition. *Prentice Hall*.

- [Eckberg, 1992]Eckberg, A. (1992) B-ISDN/ATM Traffic and Congestion Control. *IEEE Network*, September.
- [Gallassi, 1990]Gallassi, G., Rigolio, G. and Verr, L. (1990) Resource Management and Dimensioning in ATM Networks. *IEEE Network*, May.
- [Gilbert, 1991]Gilbert, H., Aboul-Magd, O. and Phung, V. (1991) Developing a Cohesive Traffic Management Strategy for ATM Networks. *IEEE Commun. Magazine*, October.
- [Habib, 1991]Habib, I. and Saadawi, T. (1991) Controlling flow and avoiding congestion in broadband networks. *IEEE Commun. Mag.*, October.
- [Händel, 1991]Händel, R. and Huber, M. (1991) Integrated Broadband Networks: An Introduction to ATM-Based Networks. *Addison-Wesley*.
- [Hong, 1991]Hong, D. and Suda, T. (1991) Congestion control and prevention in ATM networks". *IEEE Network Magazine*, July.
- [ITU1371, 1994]ITU-T (1994) Traffic Control and Congestion Control in B-ISDN. *ITU-T Recommendation I.371*.
- [Jain, 1990]Jain, R. (1990) Congestion Control in Computer networks: Issues and Trends. *IEEE Network Magazine*, May.
- [Kung, 1994]Kung, H.T., Blackwell, T. and Chapman, A. (1994) Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing. *Proceedings of SIGCOMM'94*, August 31-Sept. 2.
- [Lazar, 1991]Lazar, A. and Pacifici, G. (1991) Control of Resources in Broadband Networks with Quality of Service guarantees. *IEEE Commun. Mag.*, October.
- [Newman, 1994]Newman, P. (1994) ATM Local Area Networks. *IEEE Commun. Magazine*, Vol 32, No 3, March.
- [Ozveren, 1994]Ozveren, C., Simcoe, R. and Varghese, G. (1994) Reliable and Efficient Hop-by-Hop Flow Control. *Proceedings of SIGCOMM'94*, August 31-Sept. 2.
- [Roberts, 1991]Roberts, J. (1991) Variable Bit Rate Traffic Control in B-ISDN. *IEEE Commun. Magazine*, September.
- [Thilakam, 1992]Thilakam, R.K. and Jhunjunwala, A. (1992) The design and flow control of a high speed, integrated packet switched network. *Computer Networks and ISDN Systems*, Vol 25.
- [Vakil, 1991]Vakil, F. and Saito, H. On congestion control in ATM networks. *IEEE LTS*, August.
- [Wernik, 1992]Wernik, M., Aboul-Magd, O. and Gilbert, H. (1992) Traffic Management for B-ISDN Services. *IEEE Network*, September.
- [Yazid, 1992]Yazid, S. and Mouftah, H.T. (1992) Congestion Control Methods for B-ISDN. *IEEE Commun. Magazine*, July.

## 7 BIOGRAPHY

**Abdo Abdulmalak** received his Ph.D in telecommunication from the University Paul Sabatier of Toulouse. He is currently working as a research associate in the Department of Computer Science of the university. His research interests include computer system performance, and ATM network management. He can be reached via electronic mail at [abdulmalak@irit.fr](mailto:abdulmalak@irit.fr).

**Yves Raynaud** is a Professor at the University Paul Sabatier. He is the Founder and Director of the SIERA laboratory. His research interests include software engineering, network management, parallel processing, performance evaluation and high speed networking. He can be reached via electronic mail at [raynaud@irit.fr](mailto:raynaud@irit.fr).