ABR Congestion Control Schemes and the Multicast Problem

Muddassir TUFAIL and Bernard COUSIN
IRISA, Campus de Beaulieu
35042 Rennes Cedex (France)
{mtufail, bcousin}@irisa.fr

Abstract

Congestion control mechanisms for ABR traffic in ATM network revolve around rate-based and credit-based techniques. In rate-based techniques, resource management cells are circulated from source to destination and vice versa. These techniques have the advantage of exploiting the informations of all the nodes of a network that constitutes a particular connection. Thus, they are more flexible but less dynamic. Credit-based techniques employ hop-by-hop control and are more dynamic. They manage the buffer available at adjacent nodes. Credit-based scheme may employ adaptive buffer management which, along with the advantage of saving buffer space, have certain limitations, e.g., 1) maintaining upstream and downstream consistency and 2) calculating the buffer share for each virtual channel.

An important part of our paper is devoted to highlight the various issues in the domain of multicast ABR traffic. The behavior of multicast ABR nodes is investigated when subjected to traffic congestion. Moreover, implementations of rate-based and credit-based techniques are examined for multicast traffic session. We are of the opinion that the simple implementation of these techniques does not guarantee the maximum utilization of resources available within the network. As an important fact, we deduce that the multicast case, being more explosive and propagative as compare to point-to-point traffic, deserves special attention in order to avoid its n-fold after-effects. Finally, we present a novel congestion control scheme that, based on a priority algorithm, extends conventional congestion control schemes in order to efficiently handle multicast ABR traffic.

Keywords: rate-based schemes, credit-based schemes, multicast ABR traffic

1 Introduction

For nearly a century, the primary purpose of telecommunication networks has been to support a communication network suitable for transmitting quality voice sounds between telephones. Recently, the emergence of high speed networks (e.g. ATM) have introduced the transfer of video, graphics and data along with the sound. The evolution of digital technology has helped us to realize our dream of integrated services over the same communication media which is the main motivation behind the ATM development.
When we talk of available bit rate (ABR) traffic in ATM network then the management of available bandwidth and the congestion control methods are always the key issues of the discussion. In general, congestion arises when the incoming traffic to a specific link is more than the outgoing link capacity. The primary function of congestion control is to ensure good throughput and delay performance while maintaining a fair allocation of network resources to the users. There are two types of control functions [11].

- Under normal conditions (i.e. when no network failure occurs) functions referred to as *traffic control functions* are intended to avoid network congestion.
- Congestion may occur, e.g. because of malfunctioning of traffic control functions caused by unpredictable statistical fluctuations of traffic flows or network failures. Therefore functions referred to as *congestion control functions* are intended to react to network congestion in order to minimize its intensity.

Various rate-based and credit-based techniques have been proposed to perform the congestion control of ABR traffic. Out of them, EPRCA (Enhanced Proportional Rate Control Algorithm) presents a sophisticated rate-based control method and on the other hand, FCVC (Flow Controlled Virtual Channels) is a modern credit-based scheme. There is another famous scheme called ECCN (Enhanced Credit-based Congestion Notification) which implements rate-based and credit-based methods in parallel.
The domain of multicast traffic did not attract too much attention in past despite of its various applications e.g. data applications and other services such as LAN emulation. The congestion problem in ABR multicast traffic is much more propagative and complicated as compare to one in point-to-point traffic. The need to better utilize the network available resources is equally important in multicast sessions. The description of a possible congestion control scheme for multicast ABR traffic, in this paper, is destined to develop a priority algorithm which would be implemented in parallel with a congestion control scheme.

2 Congestion Control Schemes

This section presents a review over different congestion schemes proposed for **ABR Traffic**. Before entering into the discussion of congestion schemes we would like to define some important criterias for a good congestion scheme [3].

- A scheme should do the fair allocation of available bandwidth among contending VCs.
- A good scheme works well in LAN as well as in WAN.
- The scheme must be implementable with less possible increased complexity in switching nodes.
- In addition to congestion control, the scheme should also optimize the network utilization.
- Scheme must be robust enough to the loss of congestion control cells (credit cells, resource management (RM) cell).

2.1 Rate-Based Schemes

Many techniques have been proposed which implements a congestion control based on throughput rate of individual channels. Ramakrishnan-jain presents a technique which control the rate of a source by adjusting the window size [7]. Newman's technique uses the negative feedback resource management cells which, remarkably, reduces the bandwidth consumption as RM cells are produced only in the case of congestion [17]. At the same time, in case if the RM cell is lost the source will continue increasing rate which becomes more serious for a network already congested. This problem was resolved by Hluchyj and Yin scheme which used the positive feedback technique [18]. But this was criticized for using the bandwidth un-necessarily when the network is not congested. This was improved by Barnhart, restricting the bandwidth of RM cells to a fixed portion of the total bandwidth used by the ABR traffic. This method was named as Proportional Rate Control Algorithm (PRCA). Eventually, a more advanced version of the rate-based scheme appears in the market which calculates the explicit rate allocated to each VC and was named as Explicit rate-based control scheme by Jain and Charny. An enhanced version of explicit rate-based scheme is presented here [6].

2.1.1 Enhanced Proportional Rate Control Algorithm EPRCA

EPRCA is a synthesis of PRCA and explicit rate based control. EPRCA calculates a rate as PRCA did by using the previously allowed rate and any single feedback received from the network, but then it will equate the newly allowed rate with the minimum of this calculated rate and the most recent explicit rate (ER) received from the network. As required by the explicit rate scheme, the source, still, generates a stream of RM cells, which the destination will loop back. The source does so at a rate proportional to the allowed cell rate (ACR), and each RM cell contains a single bit congestion indicator (CI) as required by the PRCA, as well as an explicit rate (ER) field. Each switch has the option of sending the feedback using the explicit rate field, the congestion indicator or both [6, 3]. The destination monitors the EFCI bit of data cells. If the most recently seen data had EFCI bit set, then the destination node mark the CI bit in the RM cell.In brief, on the reception of a RM cell the source behaves as:

If CI=0 then New ACR=min(ACR+AIR, ER, PCR) such that New ACR $\leq$ PCR
If CI=1 then the ACR=ACR*RDF such that New ACR $\geq$ MCR

(RDF= reduction factor, AIR= additive increase to rate, PCR=peak cell rate, MCR=minimum cell rate.)
The reduction in the source rate is automatic i.e. in case, the RM cell is lost then after a certain time-out, the source will reduce its rate by RDF.
EPRCA scheme allows the rate of a source to oscillate less widely than with single bit feedback. Rate-based scheme, requiring a round trip delay between source and the destination, is not dynamic enough to reply sudden data burst arrivals. In WAN, these sudden data burst tend to neutralize each others, so rate-based scheme is preferred for WAN.

2.2 Credit-Based Schemes

Credit-based schemes control the number of cells buffered at each node traversed by a connection.The approach consists of per link, per VC, window flow control. Before forwarding any data cell over the link, the sender needs to receive credits for the VC from the receiver. At various times (say after sending N2 data cells forward), the receiver sends credits to sender indicating availability of buffer space for receiving data cells of the VC. After having received credits, the sender is eligible to forward some number of the data cells of the VC to the receiver according to the received credit information. Each time the sender forwards a data cells of a VC, it decrements its current credit balance for the VC by one [8].
Each node maintains a separate queue for each VC. If there is only one active VC, the credit must be enough to allow the whole link to be full at all times. In other words [3]:

Credit $\geq$ Link Cell Rate * Link Round Trip Propagation Delay

Flow control consortium company has proposed Quantum Flow Control (QFC) technique which performs hop-by-hop control [5]. The number of cells buffered are controlled on VC level and on link level as well. In other words, two techniques are employed in parallel here which are Flow-Controlled Virtual Channel (FCVC) and Flow-Controlled Virtual Link (FCVL). FCVC is described below. FCVL works the same way except controlling the variables are for link instead for VCs.

2.2.1 The Flow Controlled Virtual Channels (FCVC)

This is per VC credit based scheme to perform link-by-link flow control. There are three techniques for efficient memory management namely: N123, N123+ and N23, out of them N23 has gained more popularity [12]. Before entering into details of each proposal, we would like to define some notations:

- R= Round trip link delay between the current and upstream nodes, including both the link propagation delay and the time for handling data cells and processing credit cells at the two end points.
- B_{VC}= Targeted bandwidth of a VC time over R.
- B_{link}= Peak bandwidth of the underlying physical link over time R.
- Cell_size= 424, which is the number of bits in an 53-bytes ATM cell.
- N1= The N1 zone catches all in-flight data cells for the VC.
- N2= The N2 zone defines that the node is eligible to send a credit cell for a VC to the upstream node, only after it has forwarded N2 data cells of the VC to the downstream.
- N3= The N3 zone prevents data and credit underflow, so that the VC can sustain its targeted bandwidth as long as the upstream node has data to forward and the downstream node has space to receive them. For all the three proposal the value of N3 is calculated as;

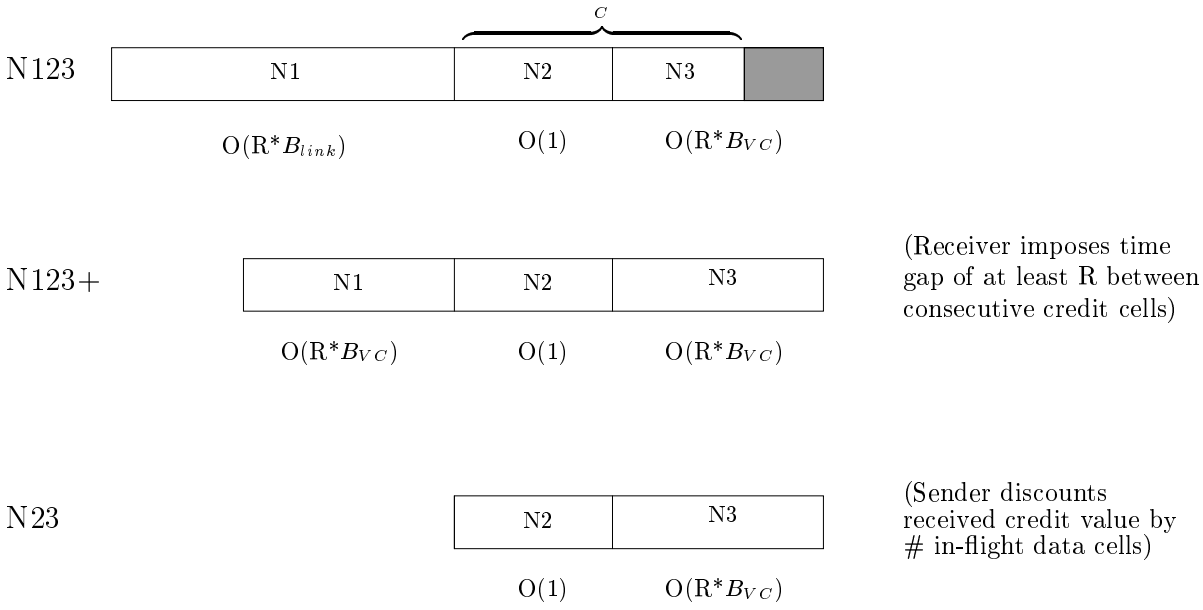

Figure 1: Three credit-based flow control schemes and their buffer sizes

$$\text{N3} = \text{R}^* B_{VC}/\text{Cell\_size}$$

1. **Basic N123 Algorithm:** The current node is eligible to send a credit cell (to the upstream node) for a VC each time after it has forwarded at least N2 data cells of the VC since the previous credit cell for the same VC was sent. The N1 value is calculated as;

   $$\text{N1} = \text{R}^* B_{link}/\text{Cell\_size}$$

   The credit cell will contains a credit value C equal to the number of the unoccupied cell slots in the combined area consisting of the N2 and N3 zones (see figure 1). A credit cell is not sent when combined area is totally occupied. The upstream node maintains a count, called credit_count, for the VC. Initially value of credit-count is set to be N2+N3. Each time the upstream node forwards a data cell of the VC , it decrements the credit-count by one. The sender stops sending when credit_count is zero. On receiving a credit cell, the value of credit_count is put equal to C.
   The N1 field in N123 algorithm assures no data overflow, as it is calculated to capture all in-flight data cells during the interval R. This scheme is robust and self-healing against any credit cell loss/error. The N3 field guarantees no data underflow as it sustains a VC's targeted bandwidth as long as there are no corrupted credits cells.

2. **The N123+ Scheme (Receiver-Enhanced, Credit-Based Flow Control Scheme):** An additional work is done at the receiver that involves imposing a time gap of at least one round-trip link delay R between the sending of two consecutive credit cells for the same VC. This modification leads us to calculate the value of N1 as;

   $$\text{N1} = \min(\text{N2+N3},\ R * B_{link}/\text{Cell\_size})$$

   Since N1$\leq$ N2+N3, the total buffer size, N1+N2+N3, for the VC is no more than 2*(N2+N3), which is independent of the peak bandwidth of the physical link. When N2+N3 is smaller than $R * B_{link}$/Cell_size, the N123+ scheme is more attractive than N123 scheme as far as minimizing the buffer is concerned.
   By imposing a gap of at least R between two consecutive credit cells, it is easy to show that the sender can forward at most 2*(N2+N3) data cells over any time interval of length R, which in other words bounds the realizable bandwidth by a VC.

3. **The N23 Scheme (Sender Enhanced, Credit-Based Flow Control Scheme):** On this scheme, we no more require N1 field in the buffer and there is no restriction of a time gap of at least the round-trip delay R between two consecutive credit cells for the same VC. For calculating the credit_count;

   $$\text{Credit\_count} = \text{Credit value in the newly received Credit cell} - \text{E}$$

   Where E is the number of data cells the sender has forwarded over the VC for the past round-trip time R.

The N23 scheme can be considered as an "ultimate credit scheme" in the sense that the sender is allowed to forward data cells against the current credit and sometimes also against a credit yet to come. It has all the properties of N123 scheme and moreover, requires lesser buffer per VC.

2.2.2 Limitations of Credit-Based Flow Control Schemes

The credit-based schemes are very dynamic as their response time is very small (node-to-node round trip delay). They are preferred for LAN as buffer are managed locally at each node. The buffer allocation per VC can be done in two ways [13].

1. **Static credit control scheme:** In case when buffer allocation per VC is done statically then the simulations results have shown that the memory requirements for even small distances are large, in the range of some megabytes per port, but not impractical. However the memory requirements for high capacity wide area links, which varies from gigabyte to tetrabyte per port, are clearly infeasible. Even if memory continues to improve exponentially, we must remember that transmission capacities are following the same trend, so do the number of VCs supported. Thus the memory requirement for the static based scheme will increase quadratically since it is linearly dependent not only on the capacity, but also on the number of VCs.
2. **Adaptive credit control scheme:** Adaptive schemes attempt to reduce the memory requirements by dynamically allocating a shared buffer among VCs at the cost of complexity and reduced performance. The difficulties encountered, when designing such scheme, can be placed in following topics:
   - **Maintaining upstream and downstream consistency:** Consider a case that the buffer allocation for each VC at a downstream node has changed. Now it must delay using any new allocation limit until it is sure that upstream node has received. This means waiting at least a propagation delay, and even longer if it includes mechanism to make scheme robust to message cell losses. Since different input links have different propagation delays, keeping track of all the allocation status is very difficult. For simplicity we take single worst case interval for all inputs links, hence performance is limited.
   - **Calculating the buffer share for each VC:** So how to divide up the buffer among the contesting VCs. In addition to calculating new buffer allocations, one must make sure that there is sufficient remaining buffer space to hold the cells already in flight, plus those that will be sent during the time that it takes:
     (a) downstream node to calculate the new allocation
     (b) upstream node to receive the new allocation
     (c) upstream node to update its parameters
   - **Dependence on sibling propagation delay information:** Operation of the adaptive scheme requires that the upstream and downstream sides of a link not only accurately know the propagation delay of the link in question, but also the propagation delay of all other sibling links. In case when a single adaptive buffer upstream side could serve multiple output ports, even more dependence are introduced. It is not hard to imagine networks where a single change could effect every node in the network and in big networks cables are plugged/unplugged and rerouted very often. So all times, a reconfiguration of whole network is required.
   - **Synchronization of upstream, downstream and sibling parameters:** The above mentioned problem requires afterwards that even after the correct delay information has received, the process running on each end of the link must synchronize. During the synchronization period the throughput and fairness are extremely poor. The synchronization period lasts for a period of about the allocation interval itself (which is likely to be in range of 20 to 100ms).

2.3 Congestion Schemes Controlling Both Rate and Credit

In order to combine the flexibility of rate with the performance of credit, following integrated approaches are considered [9]. The first two approaches are combination of rate and credit based while the third is a true integration.

2.3.1 Rate in the WAN, Credit in the LAN

The proposal is simply to use the rate based scheme in WAN and the credit based scheme in LAN. This integration suffers a major drawback of creating two types of ATM interfaces.
This interface acts as a virtual source and virtual destination, terminating the credit and rate control

loops and interconnecting them. Thus the LAN looks like a rate based source to the WAN, and the WAN looks like a credit based subnet to the LAN. The credit scheme on the LAN side is terminated by returning credit cells for VC's whose data cells are forwarded on to the WAN.

2.3.2 Rate is Default, Credit is Optional

In this solution, rate based control is required in the WAN, and it is the default scheme in the LAN. Static credit based scheme is permitted as an option within the LAN, selected on per connection basis, when a connection is established.
The virtual source/destination interface between the LAN and the WAN is still required if the credit option is selected on the local area portion of the connection.

2.3.3 One Size Fits All

The third proposal is to use an encoding in the RM cell to provide not only rate information in the cell specifying the rate a VC can flow, but also has the validity count field that is associated with rate. The validity field may be interpreted as *the number of the cells transmitted by a VC before the rate that it is currently using becomes invalid.*
This permits rate and credit based switches to be mixed within the local area without any special interface equipment. A rate switch can operate downstream from a credit switch because the RM cell contains the sum of the rate and credit control information. The source will not transmit at a rate greater than that permitted by the rate switch and will send no more cells than is permitted by the credit switch.

2.3.4 Parallel Implementation

There are schemes which implement a link-by-link credit-based flow control algorithm in parallel with an end-to-end per VC congestion notification protocol. Here is the brief description of one such scheme which is called as *Enhanced Credit-based Congestion Notification (ECCN) scheme* [10]. Effective congestion avoidance is achieved by link-by-link credit based flow control algorithm, without the need of per VC buffering at the switches. Buffer allocation in a switch is done per port/link basis. A down stream node sends credit cell to its upstream node after forwarding a certain number of cells from the corresponding input port of the downstream node. The sender keeps the record of cells sent and do not cross the limit of credit allocated which guarantees no cell loss. Forward congestion notification can be generated at switches by marking the data cells of congested VCs. Upon the receipt of first marked data cell, a destination end system sends back a congestion notification cell to the corresponding source end system. The source will slow down its sending rate by a certain factor after receiving a congestion notification cell. During the delay caused by this end-to-end feed back, the link-by-link credit control will ensure no cell loss. There are two situations in which a intermediate node will mark congestion: when a node is congested, or when a node attempts to provide fairness among VCs.
A switch constantly monitors the sending rate of each VC by counting the number of cells transmitted in some fixed time intervals. If an output port of a switch is fully utilized, the switch will compute the fair sending rate of each VC that uses the output port. The switch can then mark the data cells of one or more VCs having maximum unfair sending rates. This avoids the head of line blocking.
This techniques avoids buffering/queuing of data per VC basis which considerably reduces the switches complexity especially in case of WAN having enormous VCs. On the other hand switches take certain time before stabilizing and optimizing the network utilization. This scheme is equally applicable on both input and output buffer switches.

3 Traffic Congestion in Multicast

A lot has been said and discussed for the traffic congestion control of point-to-point ABR traffic. The domain of multicast is still thirsty despite of its bright future prospects e.g. data application and other services such as LAN emulation. The phenomena of better utilizing the network available resources is more complicated in case of multicast traffic as the behavior of a father node (B in figure 2) depends upon the collective response of his son nodes (C,D and E). More clearly, if any son node of the multicast traffic is blocked, due to any reason, then diffusion of message by the father node B to its other obedient son nodes has to be either stopped or rescheduled according to the control policy adopted. An effort has been made, in the following sections, to present possible congestion control schemes for multicast traffic.

3.1 Multicast Congestion Control by Credit-Based Technique

A credit-based technique implements a hop-by-hop control. The buffer at nodes is allocated for each VC separately. Consider the following scenario.
The node B (see figure 2) receives the message from A and copies it on three channels to C, D and E. At the same time it sends back a credit, equivalent to number of cells forwarded, for its channel to node A. The node A has a right to transmit certain number of cells to node B as dictated by the credit value. On the other hand node B waits for the credit reception from all its son nodes (C, D and E). Every thing will go smooth if it receives credit from all its son nodes **within a certain time delay** and each credit value falls **within a certain quantified range**. What can happen if network conditions changes rapidly whose probability is fairly high when we talk of data bursts arriving at any time in networks nodes? We consider the following two cases:

1. The node C is congested where as the other two son nodes D and E have sent their credits to node B. The node B is ready to send the data but cannot copy the message on the channel to node C. However, the credits received fall within a certain quantified range. In this situation, there are no many options left for node B, either it disconnects the branch to node C or it waits until node C sends the credit.
   - Considering the case when node C is disconnected. The retransmission of data, which has already been transferred to other son nodes during the period the node C was disconnected, causes an un-necessary repetition of data for nodes D and E. This solution leads to the dissipation of precious network resources which could have been, otherwise, utilized for another application.
   - The second choice is to wait for the moment when node B receives the credit from node C as well. In this case there is a high probability that the credits received earlier from nodes D and E are no more valid. So a procedure of re-confirmation of credit availability has to be defined, in this case, for the nodes declared ready previously.
2. The node B receives the credits from all its son nodes C, D and E well within a certain time delay. The credit values received from all nodes are not equal. In this case the safest way is to adopt the lowest credit received (provided lowest credit value received $\geq$ MCR declared for this ABR) and forward the cells. But this will lead to the under-utilization of bandwidth available on all the branches downstream except on the one whose credit value is chosen.

Thus the simple implementation of credit-based scheme in multicast traffic does not guarantee us the maximum utilization of network available resources.

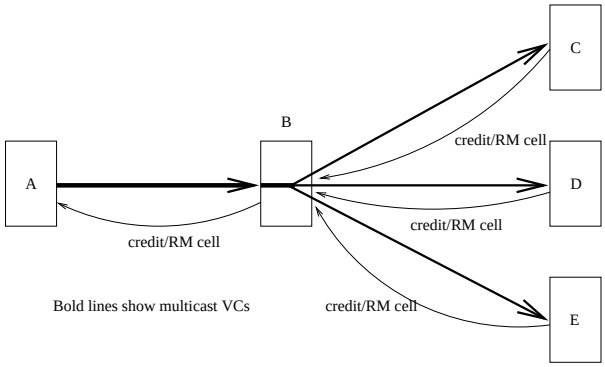

Figure 2: The multicast service in ATM

3.2 Multicast Congestion Control by Rate-based Technique

In a rate-based technique, a source transmits, periodically, the RM cells. Then, RM cells are looped back by destination node on each branch of multicast tree. Each RM cell contains a rate value agreed by all the nodes on the respective multicast branch up till the destination node. The node B (see figure 2) expects to receive RM cells **within a certain time delay** and their rate values **within a certain quantified range**. The consequences of not respecting these two constraints are same as described in section 3.1. In a multicast session, father node of the multicast tree (node B in our case) plays a vital role as it has to calculate a rate value satisfying all the downstream multicast branches. Kai-Yeung Siu has proposed multicast congestion control scheme ensuring max-min fair rate allocation [15]. It adapts the source rate to the minimum rate among all VCs to multiple destination nodes in

a multicast traffic session thus restricts the multicast flow in the branches where network resources are available.

4 A Prospective Solution of Multicast Problem

The solution we propose is a sort of parallel implementation of credit-based and rate-based schemes. Our credit-based scheme differs from the conventional one. We do not recommend credit per VC scheme for multicast.
In order to do buffer control per VC, there are different parameters to calculate (N1, N2 and N3 as suggested in FCVC method in section 2.2.1) which depend on the link speed, buffer availability at the downstream node and the distance between sender and receiver nodes. Evidently, these values would not be same for all the branches, of a multicast tree, originating from a father node. So, the father node should be tuned to which downstream branch of multicast tree? This question is difficult to answer when we apply buffer control per VC.
Our credit-based scheme implements credit per port which removes the above mentioned complexity. At the same time, to have fair allocation of buffer among the different VCs of a port, we implement end-to-end rate based scheme. At each node, the flow of each VC is observed and data cells of misbehaving VC will be marked. These marked data cells, when arrive at the destination, generate a RM cell. This RM cell dictates the source of misbehaving VC to reduce its emission rate. To tackle the problems elaborated in sections 3.1 and 3.2, we suggest to implement a priority algorithm, in ABR traffic class, which takes effect at father node. It gives priority, in certain conditions, to multicast VCs over point-to-point VCs at the node. Since the credits are per port, the priority algorithm will have the freedom to allocate buffers intelligently among different contending VCs. If the credit division results in unequal buffer allocation to different branches of multicast tree then the priority algorithm would suppress certain point-to-point VCs to satisfy the demand of multicast VCs. Consider the following case, referring to figure 2.

- The credits per port received from all three son nodes (C,D and E) are equal, but after having done the buffer allocation among multicast VCs and other VCs (which are, also, switched by the node B to downstream nodes C,D and E), we find that all branches of multicast tree are not allocated the same rates. In this case, the priority algorithm takes effect. It deprives certain point-to-point VCs of their fair share of buffer and allocate a suitable credit value to multicast VCs. This approach guarantees us the maximum utilization of network available resources.

There are many parameters to look into for the proper implementation of such priority algorithm e.g. when and whom to give priority? One thing is sure that to achieve the maximum utilization of network available resources, a trade-off between multicast VCs and point-to-point VCs is necessary.

4.1 The Priority Algorithm

This section describes the above mentioned priority algorithm. In a multicast switch, all the multicast channels have to go through a multicast scheduler before approaching the corresponding output port. At an output port, each unicast queue is assigned with a Normal Priority (NP) variable where as each multicast queue is assigned with the NP variable as well as a Multicast Priority (MP) variable. The NP and MP variables are calculated/updated at output port and at multicast scheduler respectively. The incoming cells of an input port are buffered in different queues. A queue corresponds to an output port to which are destined its cells regardless of their VCI/VPI.

At the Multicast Scheduler: The multicast scheduler creates the required number of data copies of multicast session cells and buffer them in corresponding output queues. A MP value is evaluated for each queue of a multicast session. The calculation of MP variable is very simple.

- If a queue length crosses the maxthreshold value ("a" in fig 3), it attains positive MP value.
- If a queue length is shorter by the threshold_diff value ("b" in fig 3) than the longest queue of same multicast session, it attains negative MP value.
- If a queue falls in none of above two cases, its MP value is zero.

For a multicast session, the values of max_threshold and threshold_diff parameters are selected such that:
max_threshold > (the maximum possible queue length - threshold_diff)

At the Output Port: The cells from different queues are contending for the next available cell slot. Among these queues, there may be some queues belonging to multicast sessions. The output port evaluates a NP value for each of its queues regardless of their nature (unicast or multicast). The NP variable is function of two following parameters and is updated every cell slot.

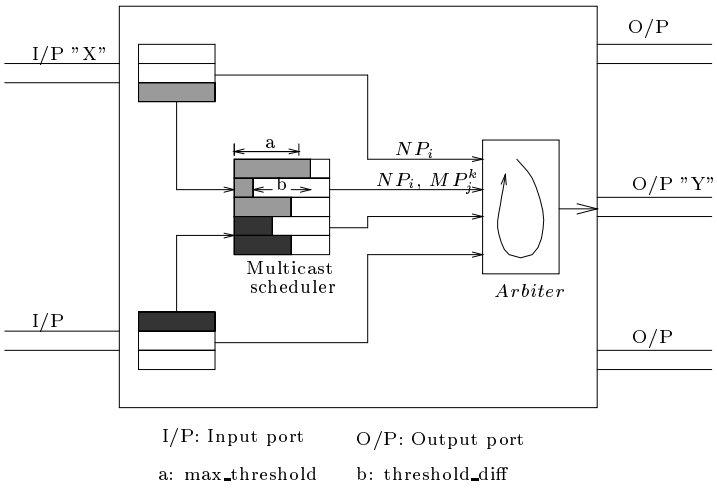

Figure 3: The implementation of priority algorithm in a multicast ATM switch

1. The percentage of buffer occupied by the queue (queue length).
2. The number of times that a cell of the queue has been refused for the available cell slot.

We describe the priority algorithm working scenario in following two categories as;

1. **Normal Condition**: When all the multicast queues at an output port have MP value zero or when there is no multicast queue at the output port, the priority algorithm picks the cell from a queue of the highest NP value and forward it in next available cell slot. The unserved queues get their NP value updated accordingly.
2. **Particular Condition**: This condition occurs whenever one or more multicast queues tend to either overfill or underfill their buffers. The algorithm looks for the multicast queue with the highest MP value and do as follows;
   (a) If the highest MP value is positive, the cell from this queue is forwarded in the next available cell slot regardless of the queue NP value.
   (b) If the highest MP value is negative, the algorithm searches for the queue with the highest NP value among the unicast queues only. A cell from this queue will be forwarded in the next cell slot.
   (c) If the highest MP value is zero, the algorithm selects a queue of highest NP value among the unicast queues along with the multicast queues with MP value zero. A cell from the selected queue is forwarded in the next cycle.

This way the priority algorithm decides dynamically the queue to be served and maintains a service balance between unicast and multicast queues. The description of priority algorithm is given in appendix A.

5 Conclusion

The paper presents an intelligent collection of selective congestion control schemes. Rate-based and credit-based schemes are discussed and finally different ways of their integration are presented. We are of the opinion that, out of four integration proposals, the parallel implementation scheme (see section 2.3.4) can open us new out-looks specially in the domain of multicast congestion control.
The multicast section of this paper brings attention to a n-fold problem. The example taken in this section, despite of its simplicity, reveals that how complex may be a congestion control in multicast? To tackle such problem, we propose to create a notion of priority for multicast VCs over point-to-point VCs which will favor multicast VCs, according to the situation at father node, while calculating the buffer allocation. This priority will, surely, hamper temporarily point-to-point VCs to emit data at their deserved share but on the other hand we get the the best utilization of network available resources which proves it as a good trade-off. The development of the priority algorithm is under study and requires to be analyzed more to cope with real life problems. We are convinced of the parallel implementation of this priority algorithm along with a congestion control scheme which, in itself, employs simultaneously both the concepts of rate-based schemes and credit-based schemes.

References

[1] Rainer Händel, Manfred N. Huber, Stefan Schröder. ATM Networks, Concept, Protocols, Applications. *Second Edition 1995, Addison-Welsey Pub. Co., pp. 1-19.*

[2] Kai-Yeung Siu, Raj Jain. A Brief Overview of ATM: Protocols Layers, LAN Emulation, and Traffic Management. *Computer Communications Review (ACM SIGCOMM), April 1995.*

[3] Raj Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey. *ATM Forum/95-0177, Oct. 1995.*

[4] Pierre Rolin. Rseaux Haut Dbit. *Edition Herms, Paris, 1995.*

[5] Mike Gaddis, Walter Kelt. Quantum Flow Control. *Version 2.0 July 1995 FCC-SPEC-95-1.*

[6] Flavio Bonomi, Kerry W. Fendick. The Rate-Based Flow Control Framework for The Available Bit Rate Service. *IEEE Network, March/April 1995.*

[7] K. K. Ramakrishnan, R. Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. *ACM Trans. Comput. Sys., vol. 8, no. 2, 1990, pp. 158-181.*

[8] H.T.Kung, Robert Morris. Credit-Based Flow Control for ATM Networks. *IEEE Network, March/April 1995.*

[9] K.K.Ramakrishnan, Peter Newman. Integration of Rate and Credit Schemes for ATM Flow Control. *IEEE Network, March/April 1995.*

[10] Hong-Yi (Henry) Tzeng, Kai-Yeung (Sunny) Siu. Enhanced Credit-Based Congestion Notification (ECCN) Flow Control for ATM Networks. *ATM Forum/94-0450, May 1994.*

[11] ITU-T Recommendation I.371. Traffic Control And Congestion Control in B-ISDN. *March 1993.*

[12] H.T.Kung Harvard University. The FCVC (Flow Controlled Virtual Channels) Proposal for ATM Networks. *Proc. International Conf. on Network Protocols, San Francisco, California, Oct. 1993, pp. 116-127, Modified April 8, 1994.*

[13] David Hughes, Pat Daltey. Limitations of Credit-Based Flow Control. *ATM Forum Technical Committee, Document Number: 94-0776, 12 September 1994.*

[14] Martin de Prycker. Asynchronous Transfer Mode solutions for broadband ISDN. *Ellis Horwood Limited, Edition 1992.*

[15] Kai-Yeung Siu, Hong-Yi Tzeng. Congestion Control for Multicast Service in ATM network.*Dept. of Electrical & Computer Engineering, University of California, Ivrine. Technical Report Number: 94-03-01.*

[16] A. Romanov. A Performance Enhancement for Packetized ABR and VBR+ Data. *AF-TM 94-0425, March 1994.*

[17] P. Newman. Traffic Management for ATM Loacl Area Networks. *IEEE Commun. Mag., vol. 32, no. 8, Aug. 1994, pp. 44-50.*

[18] M. Hluchyj, N. Yin. On Closed-loop Rate Control for ATM Networks. *Proc. INFOCOM 94, 1994, pp.99-108.*

A The Priority Algorithm Description

The description of the algorithm revolve around two sites of a multicast switch whose behaviors are explained below (see figure 3).

A.1 Output Port Behavior:

If ($credit^Y > 0$ and n > 0 and m > 0)

a) If ($\exists$ i, $1 \leq$ i $\leq$ m such that $MP_i^k \neq 0$)
   then follow case "c".
   else follow case "b".

b) NP_J = max (NP_i, $1 \leq$ i $\leq$ n)
   $credit^Y = credit^Y$ - 1 /* a cell from VC_J is forwarded downstream */
   If ($VC_J \in S^Y$(U))
   then $Forward^X = Forward^X + 1$
   If ($VC_J \in S^Y$(M))
   then $Forward_J^k = Forward_J^k + 1$
   Do i= 1 ... n
   If (i $\neq$ J) then update NP_i

c) If m > 1
MP_R^k = max (MP_i^k, 1 ≤ i ≤ m)
If m=1
R=m
If n > m
If $MP_R^k = 0$
NP_J = max {max (NP_i such that $VC_i \epsilon S^Y$(U)), max (NP_i such that $VC_i \epsilon S^Y$(M) and MP_i^k=0)}
$credit^Y = credit^Y$ - 1 /* a cell from VC_J is forwarded downstream */
If ($VC_J \epsilon S^Y$(U))
then $Forward^X = Forward^X + 1$
If ($VC_J \epsilon S^Y$(M))
then $Forward_J^k = Forward_J^k + 1$
If $MP_R^k < 0$
NP_J = max(NP_i such that $VC_i \epsilon S^Y$(U))
$credit^Y = credit^Y$ - 1 /* a cell from VC_J is forwarded downstream */
$Forward^X = Forward^X + 1$
Do i= 1 ... n
If (i ≠ J) update NP_i
If (n=m or MP_R^k >0)
$credit^Y = credit^Y$ - 1 /* a cell from VC_R is forwarded downstream */
$Forward_R^k = Forward_R^k + 1$
Do i= 1 ... n
If (i ≠ R) update NP_i
If ($credit^Y > 0$ and n > 0 and m = 0)
Follow case "b" described above

A.2 Multicast Scheduler Behavior:

If nms > 0
Do k=1 ... nms
L_H^k= max(L_i^k, 1 ≤ i ≤ $ncms_k$)
If L_H^k > max_threshold
$MP_H^k = L_H^k$ - max_threshold
Do j= 1 ... $ncms_k$
If (j ≠ H)
If ((L_H^k - L_j^k) > threshold_diff)
MP_j^k = threshold_diff - (L_H^k - L_j^k)
If (∀ i, 1 ≤ i ≤ $ncms_k$ such that $Forward_i^k \geq 1$)
then $Forward^X = Forward^X + 1$

A.3 Variables and Parameters

- MP_j^k: multicast priority defined for j^{th} queue of multicast session k.
- NP_i: normal priority defined for i^{th} queue at the output port.
- n: total number of queues present at an output port.
- m: total number of multicast queues present at an output port.
- nms: total number of multicast sessions present in the multicast scheduler.
- $ncms_k$: total number of copies of k^{th} multicast session cells, created by multicast scheduler.
- L_j^k: length of j^{th} queue of k^{th} multicast session.
- max_threshold: the maximum threshold value of queue length of a multicast session in multicast scheduler.
- threshold_diff: the threshold difference of a queue length from the longest queues of the same multicast session.
- $Forward^X$: number of cells forwarded by the port "X".
- $Forward_j^k$: number of cells forwarded by j^{th} queue of multicast session k.
- $credit^Y$: the remaining buffer of downstream node of output port "Y".
- S^Y(U): set of all the unicast queues present at the output port "Y".
- S^Y(M): set of all the multicast queues present at the output port "Y".