Performance of feedback fusion algorithms for point-to-multipoint ABR connections in a heterogeneous*flow

Mudassir Tufail
INRIA/IRISA Rennes
Campus de Beaulieu, 35042, France.

Bernard Cousin
INRIA/IRISA Rennes
Campus de Beaulieu, 35042, France.

Abstract

For ABR multicast[1] (point-to-multipoint) flow, ATM Forum [9] proposes to have fusion of feedbacks, coming from multiple destinations of the connection, at branch-points. For that matter, many fusion algorithms are proposed in the past for multicast ABR applications [3, 2, 4, 1]. In the conception of these fusion algorithms, the possible sharing of bandwidth by any unicast ABR flow on the same link has not been taken into account which makes that the fusion algorithms lack in maintaining a service balance between unicast and multicast ABR flows and eventually the optimal utilization of available bandwidth is not attained. We solve these problems by employing, at switches, an arbitration algorithm, that observes the rate at which queues, for unicast as well as for multicast connections at an output port, are served and accordingly assigns them certain priority levels. The results show that the service balance is maintained between unicast and multicast queues in accordance to their fair share and the utilization of link available bandwidth is improved.

Key words: point-to-multipoint ABR service, fusion algorithms, service balance, arbitration algorithm.

1 Introduction

The ABR service of ATM networks is inherently closed loop. The source generates a steady stream of Resource Management (RM) cells and the destination loops them back. ABR traffic control for point-to-multipoint connections entail that the source be controlled to the minimum rate supported by all the leaves (or destinations) of a multicast tree. RM cells are replicated, alike data cells, to all the branches attached to a branch-point, where a branch is defined as any point-to-point segment of the point-to-multipoint tree and a branch-point is located at the intersection of two or more branches. When these replicated RM cells are returned back by leaf nodes, there is a fusion of RM cells at branch-point(s). For the moment ATM Forum does not specify a fusion algorithm to be implemented at branch-point, though there have been a number of proposed algorithms [1, 2, 3, 4] and are presented in the next section.

1.1 Review of previous work

A fusion algorithm gathers the information carried by the BRM (Backward RM) cells received from the branches, calculates the throughput accepted by all the downstream branches (and by itself also) and finally sends upstream a BRM cell. Following are the different propositions:

1. On receiving the first FRM (Forward RM) cell, a branch-point calculates throughput and returns a BRM cell [3]. This scheme does not wait for feedback from all branches thus suffers from consolidation noise[2].

2. Kai-Yeung and Hong-Yi Tzeng describe a Multicast Extension Algorithm (MEA) meant to extend unicast rate control protocol to a multicast environment [4]. After receiving at least a BRM cell from each branch, the branch-point transmits a BRM cell to the root. MEA suffers large transient response delays but does not have consolidation noise problem.

3. The consolidation noise problem is, more or less, resolved by Wenge Ren [2] in different propositions listed below:

*Heterogeneous flow refers, here, to a multiplexed unicast and multicast (point-to-multipoint) ABR flows on a link.

[1]The term "multicast", wherever employed in this article, means point-to-multipoint.

[2]Consolidation noise is a problem where a BRM cell generated by a branch-point may not consolidate feedback from all tree branches which may, erroneously, give feedback as high as peak cell rate, if that branch-point itself is not congested [1].

(a) In one algorithm, branch-point sends a BRM cell on the reception of a FRM cell when at least one BRM has been received from a leaf. This additional condition of AtLeastOneBRM renders the scheme slower in its transient response. However, the consolidation noise problem is partially solved.

(b) The branch-point does not generate BRM cell but the BRM cell, that is received from a leaf immediately after a FRM cell has been received by the branch-point, is passed back to the source, carrying the minimum values.

(c) In another algorithm proposed by W. Ren [2] where a BRM cell is passed to the source only when BRM cells have been received from all branches.

4. Couple of algorithms are presented in [1] which are briefly described below. The authors in [1] define the overload and underload conditions of a branch. A branch is said to be overloaded if indicated feedback is *much* lesser (e.g. by a certain multiplicative factor) than the last feedback. Following are the proposed algorithms [1]:

   (a) **Fast Overload Indication algorithm:** An immediate feedback is sent to source in case of overload. In this algorithm the ratio of source generated FRM cells to BRM cells received by the source may become more than 1 which results in increasing BRM cell overhead.

   (b) **RM ratio control option algorithm:** To avoid the problem of BRM cell overhead in above algorithm, a SkipIncrease register is introduced which is incremented when ever a BRM cell is sent before a feedback, from all the branches, has been received. In case of underload condition, if the value of the SkipIncrease register is more than zero, then this particular feedback is ignored and SkipIncrease is decremented.

   (c) **Immediate rate calculation option:** This algorithm not only takes care of overload condition in a branch but also the potential overload situation at the branch-point itself.

1.1.1 Performance defaults

The different proposed fusion algorithms, for multicast ABR flow, though are getting more optimum for multicast flow but may penalize a unicast flow sharing the same link available bandwidth, refer section 3.2. The fusion algorithm 4.b, being faster, penalizes unicast ABR flow and on the other hand fusion algorithm 3.b, being simpler and relatively slow, can't get the multicast ABR flow its fair share. Fusion algorithms are found to have disturbed the service balance between unicast and multicast ABR flows if sharing the same available bandwidth.

We propose to employ, at switches, an arbitration algorithm, described below, that observes the rate at which queues, for unicast as well as multicast connections at an output port, are served and accordingly assigns them certain priority levels.

2 Arbitration algorithm

The arbitration algorithm is a service scheduler for ABR service class which selects the appropriate cell among the backlogged queues (unicast and/or multicast) at the time whenever there is a available bandwidth at the output port. A separate queue is maintained for each connection[3].

The proposed arbitration algorithm defines two variables: Normal Priority (NP) and Multicast Priority (MP). At an output port, NP is associated to all ABR queues (unicast/multicast) whereas MP is associated to multicast queues only.

2.1 Normal Priority (NP)

NP is associated to all ABR queues regardless of their nature, either unicast or multicast. NP is initially set to zero and is increased by one at the rate of queue's Mean Allowed Cell Rate (MACR) in cell/sec. The MACR value is updated at the reception of each FRM cell. Initial value of MACR is at-most ICR (Initial Cell Rate) of the source. NP value of an ABR queue is decreased by one at each service of its cells and it may attain negative value if the corresponding queue gets additional service at an output port.

At the reception of each FRM cell, MACR is updated as MACR = MACR + (ACR[4] - MACR) * AVF if either MACR > ACR with queue in congestion state or MACR * VCS < ACR with queue in non-congestion state where AVF means AVerage Factor (normally taken as $\frac{1}{16}$) and VCS is VC Separator (VCS ≤ 1) which avoids the transient increase in VC's through-

[3]The connections with same QoS (Quality of Service) requirements may be grouped in a single queue.

[4]The arriving FRM cell contains the source's Allowed Cell Rate (ACR).

put. This method is named as "735R1" in NIST ATM simulator [10] used for our simulations.

2.2 Multicast Priority (MP)

Multicast Priority (MP) is assigned to multicast ABR queues only. The switches monitor the queue lengths of multicast ABR connections at an output port and assign each of them a Multicast Priority (MP) which may be either *active*, *sleep* or *neutral*. In this regard, we define two threshold levels associated to multicast queue's lengths which are *mcast_active* and *mcast_sleep*. These two threshold levels are independent of already existing high threshold (THigh) and low threshold (TLow) queue length's level which switches, normally, employ to determine the local congestion level. Typically as the queue (unicast or multicast) increases past the TLow and crosses the THigh, congestion is declared at the switch for this particular queue. When the queue starts emptying, the condition of congestion is not removed until the queue length falls below the TLow. Initially all multicast queues have *neutral* MP value and they are updated as:

- ***Active* MP:** A multicast ABR queue whose length exceeds *mcast_active* gets itself assigned an *active* MP value.
- ***Sleep* MP:** If a multicast ABR queue length falls below the *mcast_sleep* level, then it is assigned with *sleep* MP value.
- ***Neutral* MP:** If a multicast ABR queue length falls in none of above two cases then its MP value is *neutral*.

Regardless of associated NP value, an *active* MP value renders the multicast queue at top priority for service whereas *sleep* MP value makes its at lowest priority. The *neutral* MP value of a multicast queue does not influence at all and the cells are selected according to their respective queue's NP values. Since a *sleep* MP value means that the concerned multicast queue is to be served at lowest priority i.e. it is not be served as long as there are cells in other queues (unicast/multicast) on this output port so there is time limit up to which a multicast queue can stay with *sleep* MP value. This time limit is determined by the frequency of FRM cells i.e. Nrm. If a multicast queue gets a *sleep* MP value, for example at a instant when a FRM cell arrives at the queue, and stays with the *sleep* MP value (i.e. multicast queue length could not exceed *mcast_sleep* level) till the arrival of next FRM cell then multicast queue MP value is changed to *neutral*. This ensures that cells may not stay longer in a multicast queue.

2.3 Working principles

The arbitration algorithm proceeds as follows whenever it has to select a cell, at an output port, among the backlogged ABR queues.

1. Multicast queue(s) with *active* MP value(s) is/are present, if NO then go to step 2, if YES then proceed as:
   (a) If there is unicast queue with NP value, larger than that/those of multicast queue(s), and has queue length greater than THigh then serve the unicast queue. Go to step 4.
   (b) If the largest NP value, among those of backlogged queues (unicast/multicast), is that of a unicast queue then increase its NP value by one.
   (c) If there are more than one multicast queue with *active* MP value then serve the multicast queue with NP value larger than that/those of remaining multicast queue(s) with *active* MP value(s). Go to step 4.
   (d) Serve the multicast queue, having *active* MP value and go to step 4.
2. Multicast queue(s) with *neutral* MP value(s) is/are present, if NO then go to step 3, if YES then proceed as:
   (a) Excluding multicast queue(s) with *sleep* MP value(s), if any, serve the queue (unicast or multicast) having the largest NP value. Go to step 4.
3. Multicast queue(s), if any, is/are assigned with *sleep* MP value(s). Proceed as:
   (a) If there is/are unicast backlogged queue(s) then:
      - Serve the unicast queue having the largest NP value among those of unicast queues.
      - If the largest NP value, among those of backlogged queues (unicast/multicast), is that of a multicast queue then increase its NP value by one. Go to step 4.

(b) Else serve the multicast queue with the largest NP value among those of backlogged queues. Go to step 4.

4. Decrease the queue's NP value by one.

5. Go to step 1 for next available cell slot.

3 Simulation

The proposed arbitration algorithm is tested on NIST ATM network simulator version 1.1 [10]. The available version of simulator provided a testbed for studying and evaluating congestion control mechanism for point-to-point ABR connections. This was modified and enhanced to accommodate the point-to-multipoint ABR connections with an option of two methods of congestion control techniques: one with fusion algorithm 3.b and other with algorithm 4.b.

3.1 Testbed architecture

The testbed used has 1 unicast ABR application (sabru), 1 multicast ABR application (sabrm) with two destinations (dabrm1, dabrm2) and 1 VBR application source (svbr) and is shown in figure 1. The prefix "s" stands for source and suffix "u" and "m", for ABR applications, mean unicast and multicast whereas prefix "d" means the respective destination. The host components in figure 1 signifies a work station or B-ISDN node where the applications reside. All ATM switches are enumerated with symbol "sw". The VBR application's throughput follows Poisson law which makes the leftover bandwidth vary significantly. All applications have 40 Mbit data to send. All links are at 100 Mbit/s except link3 which is at 155.52 Mbit/s. The link3 is shared by all three application's data cells so faces congestion at times. Hence, the role of arbitration algorithm at output port, serving link3, of switch "sw2" becomes important and effects queue service orders of the unicast as well as multicast applications.

3.2 Results

There are two important observations which we have considered for analyzing the performance of a given fusion algorithm with and without the arbitration algorithm. These are throughput variation and data transfer capacity. Note that the result curves, in this section, are displayed in matrix format. Thus a figure referred as A.(x,y) means the result curve at x^{th} row and y^{th} column of figure A. Moreover, left hand column of a figure (except for figure 3) shows the result curves with fusion algorithm 3.b and those at right hand column are with fusion algorithm 4.b.

3.2.1 Throughput variation

In order to measure the throughput variation of an ABR application we have noted the attached link utilization.
Refer to figure 2 which shows the multicast ABR source "sabrm" throughput variation (as determined by the link7 utilization level at different instants) under different test configurations.
The frequently varying throughput of multicast ABR source, with fusion algorithm 3.b figure 2.(1,1), is rather stable when the arbitration algorithm is applied with *mcast_active*=17 and *mcast_sleep*=7 figure 2.(2,1).
Note that the role of arbitration algorithm is more important in case fusion algorithm 3.b is applied. The average throughput in figure 2.(2,1) is 55.78 Mbit/s whereas that with no arbitration algorithm is 53.5472 Mbit/s, refer figure 2.(1,1), thus a gain of 4.2% for multicast application throughput when the arbitration algorithm is applied.

3.2.2 Data transfer capacity

The second important property to be observed is the data transfer capacity of an application. The data transfer capacity means that how many cells the application could transmit under a given algorithmic configuration[5]. The cells reception times at host3 (for multicast application) and host7 (for unicast application) are noted and compared in figure 3. The notation "x-no-arb" means the cell received at host "x" when the arbitration algorithm is not applied whereas "x-arb" signifies the cell received at host "x" when the arbitration algorithm is applied. The figure 3.(1,1) shows that there are more cells transmitted for unicast ABR application than those for multicast application when the fusion algorithm is 3.b. This phenomenon is reversed when the fusion algorithm is 4.b that is to say more multicast cells transmitted than unicast cells[6], figure 3.(2,1). This explains the need of the arbitration algorithm which could recover the service balance between unicast and multicast ABR flows. There is an

[5] Algorithmic configuration means selection of fusion algorithm with or without arbitration algorithm.

[6] Recall that both unicast and multicast applications have same amount of data to be transmitted and they both have same emission parameters i.e. PCR, ICR, MCR etc.

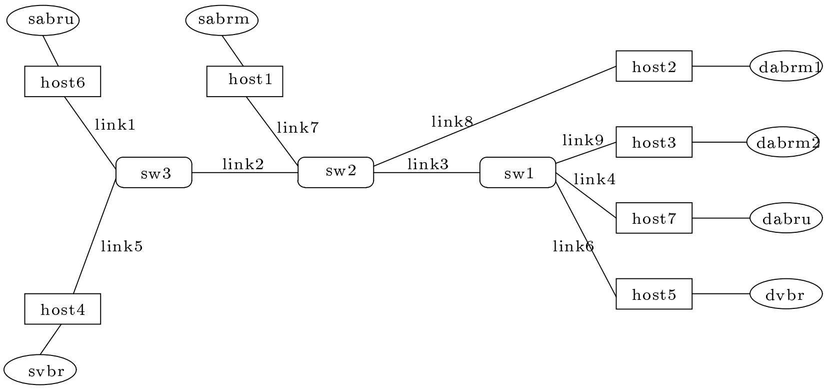

Figure 1: The testbed architecture

Figure 2: Throughput variation of sabrm source

approximate increase of 1.4% in total number of cell transferred with arbitration algorithm when the fusion algorithm is 3.b whereas for the fusion algorithm 4.b the improvement is of the order of 0.6%.

3.3 Remarks

The fusion algorithm 4.b is more complex than fusion algorithm 3.b as it maintains additional registers like LastER, NumberOfBranches, NumberOfBRMsReceived and SkipIncrease which helps it to respond fast in case of congestion and the multicast ABR source modifies its throughput accordingly. Doing this, we have noticed that fusion algorithm 4.b penalizes unicast ABR flow. On the other hand fusion algorithm 3.b, being simpler and relatively slow, can't get the multicast ABR flow its fair share. The combined performance of fusion algorithm 3.b and the arbitration algorithm is comparable to that of fusion algorithm 4.b. Moreover, we attain the service balance (between unicast and multicast ABR flows) in this way which either fusion algorithm can not ensure if implemented without the arbitration algorithm. We, therefore, suggest that fusion algorithm 3.b be implemented along with the proposed arbitration algorithm.

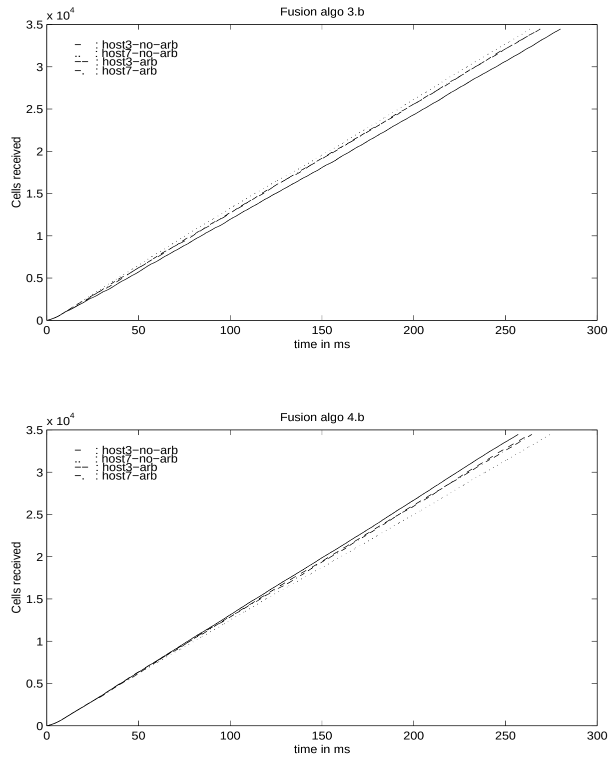

Figure 3: Number of cell received at destination

4 Conclusion

The evolution of fusion algorithms, proposed for ABR point-to-multipoint connections, has, though, optimized the multicast ABR flow, but at the same time a performance degradation of unicast ABR flow, if sharing the same available bandwidth, has also been observed. They lack in maintaining a service balance between unicast and multicast ABR flows. In this context, we propose an arbitration algorithm which ensures the required service balance without degrading the either ABR flow (unicast/multicast) properties like throughput variation and data transfer. The arbitration algorithm assigns certain priority levels (NP and MP) which help it to select an ABR queue for service at a given time. NP ensures that the ABR queue (unicast or multicast) gets its fair share at all time whereas MP increases or decreases the priority of a multicast queue with respect to its congestion level. Performance analysis of a fusion algorithm with and without the arbitration algorithm has been carried out via an updated version of NIST ATM simulator [10]. The results show that, regardless of type of the fusion algorithm used, the arbitration algorithm, in addition to the service balance guarantee, increases significantly the link available bandwidth utilization which in turn increases the total amount of data transmitted for an ABR application (unicast or multicast) within a given time interval.

References

[1] Sonia Fahmy, Raj Jain, Sivkumar Kalyanaraman, Rohit Goyal, Bobby Vandalore, Sastri Kota and Pradeep Samudra. Feedback consolidation algorithm for ABR point-to-multipoint connections. *ATM Forum Contributions 97-0615, July 1997 and IEEE INFOCOM'98 March 1998.*

[2] Wenge Ren, kai-Yeung Siu and Hiroshi Suzuki. On the performance of congestion control algorithms for multicast ABR service in ATM. *Proceedings of IEEE ATM'96 Workshop, San Francisco, August 1996.*

[3] Lawrence Roberts. Rate based algorithm for point to multipoint ABR service. *ATM Forum Contribution 94-0772R1, November 1994.*

[4] Kai-Yeung Siu and Hong-Yi Tzeng. Congestion control for multicast service in ATM networks. *Proceedings of the IEEE GLOBECOM, vol.1. pp. 310-314, 1995.*

[5] Traffic Management Specification version 4.0, *The ATM Forum Technical Committee, June, 1995.*

[6] M. Tufail and B. Cousin. Timer Imposed and Priority Supported (TIPS) Congestion Control Scheme for Point-to-Multipoint Connections in ATM. *Proc. of* 11^{th} *Congrs De Nouvelles Architectures pour les Communications (DNAC) Large Bande: Internet ou RNIS, France, 1996, pp. 65-79.*

[7] M. Tufail and B. Cousin. Proposing a new queue service scheme for ABR multicast flow. *Proc. Parallel and Distributed Computing and Networks (PDCN'97) Conference, organized by The International Association of Science and Technology for Development (IASTED) and co-operating society-IEEE Singapore. Singapore, August 11-13 1997.*

[8] Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy and Seong-cheol Kim. Use-it or Lose-it policies for the Available Bit Rate (ABR) service in ATM networks. *Journal of Computer Networks and ISDN system, January 1997, pp. 25.*

[9] Traffic Management Specification version 4.0, *The ATM Forum Technical Committee, April, 1996.*

[10] Nada Golmie, Alfred Koenig, David Su. The NIST ATM Network Simulator. *Computer Science Laboratory, Gaithersburg, MD 20899, Aug 1995.*

[11] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy and Ram Viswanathan. ERICA switch algorithm: A complete description. *ATM Forum contribution 96-1172, August 1996.*