IMPROVED ALGORITHM FOR FEEDBACK CONSOLIDATION IN MULTICAST ABR CONNECTIONS

Tamer M. El-Sayed, Mohamed N. El-Derini, Mohamed S. Selim and Magdi A. Ahmed
Department of Computer Science and Automatic Control
Faculty of Engineering, University of Alexandria, Egypt
{telsayed@contact.com.eg, elderini@iee.org, magdy@bau.edu.lb}

Abstract

A number of algorithms has been proposed for extending ABR congestion avoidance algorithms in ATM networks to perform feedback consolidation at the branch points. Schemes that attempt to maximize accuracy of feedback information tend to be slow in providing feedback to the source when the conditions in the network change. Accuracy can be traded for speed by having a switch generate feedback information before it has all the necessary information from downstream paths. This can cause consolidation noise, leading to a heavy rate at the source. In this paper, we propose improved algorithm for feedback consolidation, which combines benefits from the previous algorithms with reduced overhead. The performance of the proposed algorithm and a set of previously developed algorithms are compared under a variety of conditions. Results show that the proposed algorithm exhibits a fast transient response with accurate feedback and doesn't suffer from consolidation noise.

Keywords: ATM Networks, ABR service, congestion control, traffic management, multipoint communication.

1 Introduction

The Available Bit Rate (ABR) service of ATM Networks has been developed to support data applications over ATM networks. ABR is unique because the network switches can indicate to the source the rates at which they should be transmitting, thus avoiding congestion and efficiently utilizing network resources. ATM switches use their current load information to calculate the allowable rates for the sources. These rates are sent to the sources as feedback via *resource management* (RM) cells, which are generated by the sources and travel along the data path to the destination end systems. The destinations simply return the RM cells to the sources.

The point-to-multipoint ABR service within ATM is important for many emerging data applications. Examples of multipoint applications include distance learning, server and replicated database synchronization, advertising, searching and data distribution applications. ABR traffic management for point-to-multipoint connections controls the source rate to the minimum rate supported by all the branches of the multicast tree. Feedback consolidation at the branch points becomes a necessary operation to avoid the *feedback implosion* problem, where the number of backward resource management (BRM) cells received by the source is proportional to the number of leaves in the multicast tree. In addition, the allowed rate of the source should not fluctuate due to the varying feedback received from different leaves. A serious problem may also arise if the BRM sent by the branch point towards the root doesn't consolidate feedback information from all branches. This will introduce noise called "*consolidation noise*" leading to rate oscillation at the root [1].

A number of consolidation algorithms has been proposed in [2,3,4,5,6,7,8]. Several design and implementation considerations come into play when developing a consolidation algorithm. The oscillations and transient response of the algorithm are important. The algorithm should preserve the efficiency and fairness properties of the rate allocation schemes employed in the network switches. It must also be scalable to very large multicast trees. The implementation complexity, feedback delay, and the overhead of the BRM cells should not be increased with the increase of the number of levels or leaves of the multicast tree. Handling non-responsive branches is necessary such that they neither halt the consolidation operation nor cause overload (or underload).

Among all previously proposed consolidation schemes, "the wait-for-all" scheme (presented in [4]) avoids consolidation noise by collecting feedback news from all branches before sending a BRM to the source, but it suffers from a slow transient response. Fahmy et al. had proposed an algorithm that returns back a BRM cell not only when news are heard from all branches but also whenever a severe overload occurs [5]. A severe overload can be indicated either by a potential overload situation at the switch itself or by a feedback received from an overloaded branch. Any sent BRM cell before feedback from all the branches has been received was counted as an extra BRM cell. A technique was presented to control those extra BRM cells and maintain the RM ratio (ratio between BRM cells received and Forward RM "FRM" cells sent) at the source. The overload indication was detected using a threshold value, which is tricky to determine. The higher the threshold is, the faster the transient response is, and the higher the overhead is. While in case of lower threshold, the algorithm transient response degrades and behaves as the "wait-for-all" algorithm. Chen et al. had alleviated the threshold problem by introducing a new probability function to send an extra BRM cell, which provides more flexibility to span the speed-overhead spectrum [7]. But the algorithm lacked two important issues. Firstly, it didn't account for the local switch congestion state. Secondly, it had no technique to control the RM ratio, which may exceed one if many extra BRM cells were sent. Both issues were the main features of Fahmy et al. algorithm.

In this paper, we propose an improved algorithm for feedback consolidation that takes advantages from the previous developed algorithms with reduced overhead. The new algorithm considers the local switch congestion as a flag for sending overload feedback. But this check is only done by the switch one time per received FRM thus reducing the check overhead compared to Fahmy et al. algorithm. It depends also on the probability scheme presented by Chen et al. to send more BRM cells in case of moderate overload situations. The performance of the proposed algorithm and a set of previously presented algorithms are compared under a variety of conditions. Results of the simulation experiments indicate that the algorithm we propose doesn't suffer from the consolidation noise, while exhibiting a fast transient response with accurate feedback information using either low or high threshold.

The remainder of the paper is organized as follows. The next section provides a summary of the previous work on feedback consolidation in point-to-multipoint ABR connections. The proposed algorithm is presented in section 3 with its performance analysis and results in section 4. Section 5 discusses the tradeoffs between the compared algorithms. Concluding remarks follow in section 6.

2 Related Work

The main idea of Robert's algorithm [2] is that BRM cells are returned from the branch point when FRM cells are received, and the BRM cells contain the minimum of the values indicated by the BRM cells received from the branches after the last BRM cell was sent. This algorithm suffers from the consolidation noise problem. The method is also complex to implement because the switch has to turn around the RM cells. Most studies argue that turning around RM cells has a high implementation complexity [5].

Tzeng and Siu have presented a slightly more conservative algorithm in [3]. In this algorithm, a switch will return a BRM cell to its upstream node only when it receives an FRM cell and it has received at least one BRM cell from a downstream node since the last time a BRM cell was delivered upstream at this switch. Using this algorithm, consolidation noise can be reduced somewhat. However, the response time is slower than in Robert's scheme and the algorithm is complex to implement for the same reason of Robert's.

Ren et al. proposed an algorithm in which switches do not initiate generation of BRM cells, but instead simply forward selected BRM cells received from downstream, after modifying the contents. For example, Algorithm 3 in [4] will forward the first BRM cell received after an FRM cell has been received. The explicit rate reported in this cell will be the minimum of the peak cell rate (PCR) and all explicit rates reported to the switch since the last BRM cell was forwarded. The behavior of this algorithm is similar to Tzeng and Siu's algorithm with reduced implementation complexity, and the problem of consolidation noise still exists.

A natural extension to the previous algorithm is for a switch to return a BRM cell only after receiving feedback information from all its branches since the last BRM cell was sent (algorithm 4 in [4]). The content of this BRM cell is the minimum explicit rate reported in the BRM cells received from the downstream branches since the last BRM cell was sent. This extension, which is called (wait-for-all algorithm), can easily eliminate the consolidation noise, but the resulting slow transient response may not be satisfactory.

The main idea behind the three algorithms presented by Fahmy et al. in [5] is that the slow transient response problem should be avoided when a severe overload situation has been detected. In this case, there is no need to wait for feedback from all the branches, and the overload should be immediately indicated to the source. Overload is detected here when the feedback to be indicated is *much less* than the last feedback returned by the branch point. The "much less" condition is tested using a multiplicative factor, Threshold. The local switch congestion is considered in the third algorithm. The main disadvantage of Fahmy et al. algorithms is that they are all threshold-very-sensitive. If the threshold value is low, they exhibit the same slow transient response as the wait-for-all technique. They are also lacking a technique for handling non-responsive branches.

Ammar et al. presented a probabilistic consolidation algorithm that can be tuned using probability parameters to span the speed-accuracy spectrum [6]. For each FRM cell, extra BRM cells *may be* sent upstream during the consolidation procedure so that any detected change in conditions can be passed back to the source more quickly. The value of probability *p* (that an overload feedback is sent) can be adjusted to navigate the speed-accuracy tradeoff. The algorithm suffers from the feedback implosion problem because extra BRM cells are sent and no handling is presented to balance the FRM to BRM ratio. The local congestion state is also not considered and the algorithm is missing a technique to handle non-responsive branches.

Chen et al. designed a new algorithm to alleviate the threshold problem of Fahmy et al. algorithms by providing more flexibility to span the speed-overhead spectrum [7]. Let μ denotes the ratio of the current minimum explicit rate "MER" and the rate indicated in the last returned BRM cell. To send an extra BRM cell, Fahmy et al. presented a step probability function to send an extra BRM cell as shown in figure 1(a) while Ammar et al. presented a fixed probability function with value P_k over interval (0,1) as shown in figure 1(b). The new scheme sends an extra BRM cell with a probability *p*, which is a function of the current collected MER and the last returned feedback as shown in figure 1(c). The algorithm solved the threshold problem but is missing a technique to control the BRM to FRM ratio thus it may suffer from the feedback implosion problem. The local congestion state is also not considered.

Ros et al. proposed a new algorithm, which keeps track of the *M* smallest available rates from the branches at each branching point [8]. For this, the switch stores a matrix (called BBM matrix) with *M* entries including both the identifier of the branch (ID) and the last available rate received at the branching node. The size *M* can be adjusted depending on the scalability and performance requirements. A new feedback is stored in this structure only if its available rate is smaller than any of the available rates in the BBM matrix. A branching switch may send a BRM cell back to the root whenever a new bottleneck appears or the number of feedback values received since the last time a BRM cell was sent is equal to the number of branches. Ros et al. argued that the algorithm has zero response delay, noise stability, and small probability of noise. However, it is clear that this solution is the most expensive one. Even if just small storage is needed, it is still more than one value (MER) used by the most previous algorithms. It also costs time to maintain the BBM matrix. The algorithm also needs to handle the non-responsive branches. It would be stuck at an old feedback of a left bottleneck leaf. The local congestion state should be also considered for that algorithm to be an ideal one.

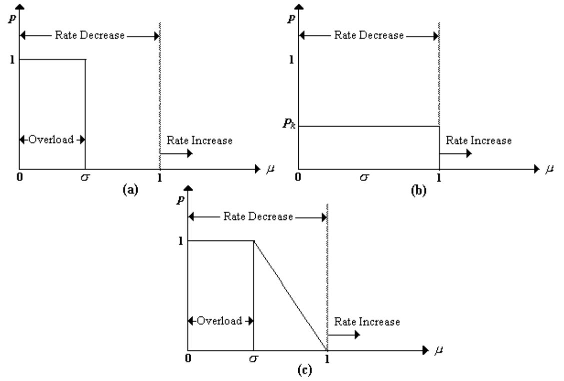

Figure 1. The probability *p* function to send an extra BRM cell: (a) Fahmy et al. Alg., (b) Ammar et al. Alg., and (c) Chen et al. Alg.

3 The Proposed Algorithm

Here, we present the new proposed consolidation algorithm that takes advantages of the previously developed algorithms. In the new algorithm, there are two ways for the switch to trigger sending a BRM cell: the first, when feedback is received from all branches, thus preserving the advantage of the "wait-for-all" scheme (accurate feedback) and the second, when either the switch itself is overloaded or a feedback indicating overload is received from a branch. To alleviate the threshold-sensitivity problem, the overload is checked using the probability function proposed by Chen et al and shown in figure 1(c). μ denotes the ratio of the current MER and the rate indicated in the last returned BRM cell while σ denotes the overload threshold. The proposed algorithm sends an extra BRM cell with a probability *p*, which is a function of the current collected MER and the last returned feedback. An extra BRM cell is sent immediately if a severe overload condition is detected ($\mu < \sigma$).

The probability *p* to send an extra BRM cell when moderate overload ($\sigma \leq \mu \leq 1$) is a linear function between two ends: one is p=1 when $\mu = \sigma$ and another is p=0 when μ=1. That is, p=(1-μ)/(1-σ) as shown in figure 1(c). Note that when a BRM cell is returned due to overload detection (severe or moderate) before feedback has been received from all branches, the counter and the register values are not reset.

Since the algorithm sends the extra BRM cells with a probability *p* if $\sigma \leq \mu < 1$ ($0 < p < 1$), the source may decrease its rate gracefully. It is beneficial to the video networking applications where large rate decrease will cause video frame rate or video quality being reduced rapidly, thereupon the resulting perceptual medium quality may not be acceptable to the users [7]. To achieve a graceful rate decrease, the Fahmy and Ammar et al. algorithms have to set the threshold (or probability) close to 1, whereas our algorithm provides adaptability in determining the threshold to decrease the rate gracefully.

In Fahmy's algorithm, the rate allocation algorithm is performed whenever a BRM is received, and not just when a BRM is being sent. Doing that, however, may involve some additional complexity. The proposed algorithm also accounts for the potential overload situation at the branch point itself, but this congestion check is only performed when receiving the first BRM cell after the last received FRM cell. This is sufficient for two reasons. Firstly, in the steady state, the rate allocations tend to be stabilized. Secondly, the new FRM cell may carry a new current cell rate "CCR" value, which means a new rate allocation. This modification decreases the overhead (of the local situation check) by a factor of N, where N is the number of branches (because N BRM cells are returned for each FRM cell), while preserving the right of the switch congestion situation to share the decision of returning overload feedback.

An RM ratio control method is necessary here to ensure that the ratio of the BRM cells, received by the source, to the FRM, sent by the same source, doesn't exceed one. We control this ratio using a variable called "SkipIncrease". SkipIncrease is incremented whenever a BRM cell is sent before feedback from all the branches has been received. When feedback from all leaves indicates underload, and the value of the SkipIncrease register is greater than zero, this particular feedback can be ignored and SkipIncrease decremented.

```
Upon the receipt of an FRM cell:
1. Multicast FRM cell to all participating branches
2. Let AtLeastOneFRM = 1

Upon the receipt of a BRM cell from branch i:
1. Let Reset = 1, SendBRM = 0
2. IF (NOT BRMReceived$_i$) THEN     // Set the flag of branch i
            A. Let BRMReceived$_i$ = 1, NBRMsRecv = NBRMsRecv + 1
3. Let MER = min (MER, ER from BRM cell)
4. IF (AtLeastOneFRM) THEN       // Check local congestion
            A. Let MER = min (MER, minimum ER calculated by rate allocation
            algorithm for all branches)
            B. Let AtLeastOneFRM = 0
5. IF ((MER $\geq$ LastER)AND(SkipIncrease > 0)AND(NBRMsRecv = Nbranches)) THEN
                                       // An underload situation
            A. Let SkipIncrease = SkipIncrease – 1, NBRMsRecv = 0
            B. Let BRMReceived = 0 FOR all branches
     ELSE IF (NBRMsRecv = NBranches) THEN // Feedback is synchronized
       A. Let SendBRM = 1
   ELSE IF (MER < $\sigma$ * LastER) THEN   // Severe Overload
       A. Let Reset = 0, SendBRM = 1
       B. Let SkipIncrease = SkipIncrease + 1
   ELSE IF (MER<LastER) THEN // Moderate Overload
            1st.  Let $\mu$ = MER/LastER
            2nd.  Let $p$ = (1- $\mu$ )/(1- $\sigma$ )
            3rd.  IF (RandomValue < $p$) THEN
                        1. Let Reset = 0, SendBRM = 1
                        2. Let SkipIncrease = SkipIncrease + 1
6. IF (SendBRM) THEN
            A. Pass the BRM with ER = MER to the source
            B. IF (Reset) THEN
                        1. Let MER = PCR, NBRMsRecv = 0
                        2. Let BRMReceived = 0 FOR all branches

When a BRM is to be scheduled:
1.Let ER=min(ER,ER calculated by rate allocation algorithm for all branches)
2.Let LastER = ER
```

Figure 2. The Proposed Algorithm

The proposed algorithm has four main features. Firstly, it gives accurate feedback due to feedback synchronization. Secondly, it takes the congestion state of local switch into consideration *with reduced overhead*. Thirdly, it detects the *severe overload* situations and sends immediate rate decrease feedback to the source leading to a fast overload transient response. While in case of *moderate overload* situations, there is a chance to send an extra BRM cell, hence alleviating the threshold-sensitivity problem. Finally, it controls the RM ratio at the source to ensure that it doesn't exceed one.

Five registers: MER, NBRMsRecv, NBranches, SkipIncrease, as well as LastER and N flags are maintained for each multipoint VC. MER stores the minimum of explicit rate (ER) values. NBRMsRecv is used to count the number of branches from which BRM cells were received at the switch after the last BRM cell was passed by the switch. NBranches stores the number of branches of the point-to-multipoint VC at this switch. LastER stores the ER value of the last sent feedback. Also, a flag, BRMReceived is needed for each branch to indicate whether a BRM cell has been received from this particular branch, after the last BRM cell was passed. The flag is stored for each output port and not for each VC, since it is needed for each branch. A Boolean flag, AtLeastOneFRM, indicates that at least one FRM cell has been received by the branch point since the last local congestion check. Four temporary variables: μ, σ, SendBRM and Reset are used. μ and σ are used to represent the ratio and overload threshold as described above. SendBRM is set only if a BRM cell is to be passed to the source by the branch point. Reset is false only if a BRM cell is being used to indicate overload conditions, and hence the register values should not be reset.

Several algorithms have been proposed to calculate the explicit rate in RM cells based on load at each port, for example, ERICA [9]. We do not assume that a specific rate allocation algorithm is used in the switch. The algorithm operates as shown in figure 2.

4 Performance Results & Analysis

This section provides a limited set of results, obtained using simulation. These results compare the performance of our proposed algorithm to three previously developed algorithms: Fahmy et al., Chen et al. and Ros et al. algorithms. A graph is plotted for the allowed cell rate of the point-to-multipoint ABR source for each algorithm. We simulate the four algorithms with two different thresholds: low (0.05), and high (0.95) for each configuration. Note that Ros et al. algorithm is threshold-independent, but we will repeat its graphs in both cases for comparison simplicity. The parameter M in Ros et al. algorithm is set to the minimum of 2 and number of branches of the multicast connection at the switch. All other parameter settings are same as indicated in [5].

4.1 Parking Lot Configuration

This configuration is shown in figure 3. It has one ABR multicast connection (from S1 to dS1, dS2... and dS5), one ABR unicast connection from SA to dSA and one CBR connection from SB to dSB. Both the ABR multicast connection and the ABR unicast connection are active from 0 to 200 ms. The CBR unicast connection is only active from 100 ms to 200 ms and the source rate is 90 Mbps. The receiver dS5 is active in the multicast from 100 ms to 200 ms. Therefore, there are two phases in this configuration: (1) Phase 1: 0 ms to 100 ms, and (2) Phase 2: 100 ms to 200 ms. The bottleneck link for the multicast connection is the link between switches SW3 to SW4. The multicast connection shares the link with the ABR unicast connection, thus the bottleneck rate is about 70 Mbps before 100 ms. The rate decrease ratio, μ, is 0.5 for the multicast connection. Figures 4 and 5 illustrate the allowed cell rate in Mbps in this configuration using high and low thresholds respectively.

The "wait-for-all" algorithm will converge in 51 ms since this is the round-trip time from the source to the farthest receiver dS1. At 100 ms, the CBR connection starts to send cells. The bottleneck link stays unchanged, but the bottleneck rate is decreased to about 29.5 Mbps. The rate decrease ratio, μ, is about 0.42 for the multicast connection. The "wait-for-all" algorithm will converge in 150.5 ms since this is the round-trip time from the source to the new joining receiver dS5.

In case of high threshold, the ACR graphs for Fahmy and the proposed algorithms indicate the same fast transient response during both phases due to the periodic switch congestion check. Both of them converge in 6 ms, since this is the round-trip time from the source to the nearest destination of the bottleneck switch. While Chen et al. algorithm waits for the round-trip time from the source to dS4 (16 ms) to detect the rate decrease in absence of the switch situation check. Ros algorithm exhibits also fast transient response since each switch stores the 2 most bottleneck values and check the switch situation at most every N BRM cells received. It suffers from some noise due to bottleneck value change. Note that the other algorithms store only the minimum rate until the feedback is received from all the branches at the time the rate allocation algorithm has been converged to the optimal value. All the algorithms detect the second rate drop in phase 2 quickly because the feedback news becomes available in the network.

In case of low threshold, performance of Fahmy algorithm degrades to the "wait-for-all" algorithm (slow transient response) since the threshold is very low and there is no chance to send an extra BRM cell even if the switch congestion is checked. Thus, the only way to send is to wait for collecting news from all branches at each branch point. Performance of Chen and the proposed algorithms here are similar to their corresponding with high threshold. This is due to the chances obtained to pass the overload feedback messages (the probability to send BRM cell in the first rate drop is about 0.53).

4.2 Chain Configuration

The chain configuration, illustrated in figure 6 consists of a point-to-multipoint connection (S1 to dS1, dS2 and dS3) where one of the links on the route to the farthest leaf is the bottleneck link (shared by the point-to-point connection SA to dSA). Also the link lengths increase by an order of magnitude in each of the last two hops (all links from the end system to the switches are 50 km). Switch 3 is the bottleneck here as the link connecting SW3 to SW4 is the bottleneck link. The bottleneck rate is 70 Mbps and the rate decrease factor is 0.5. Fahmy et al. argued that his configuration is an ideal configuration for illustrating the consolidation noise problem [5]. Figures 7 and 8 illustrate the performance of the four algorithms with high and low thresholds respectively.

In case of high threshold, Fahmy's and the proposed algorithms yield optimal performance since SW3 passes the first BRM cell (received from dS2) towards the source and doesn't needlessly wait for the BRM from SW4. Thus, the feedback is received by the source in 6.5 ms (the round-trip from S1 to dS2). Chen algorithm suffers from slow transient response. The rate of S1 drops only after 56.5 ms because SW3 must wait for a BRM cell from SW4.

The first rate decrease indicated in ACR graph of Ros algorithm is not detected by other algorithms since the rate decrease ratio is 1. This drop appears because ICR here is 150 (not 140) Mbps.

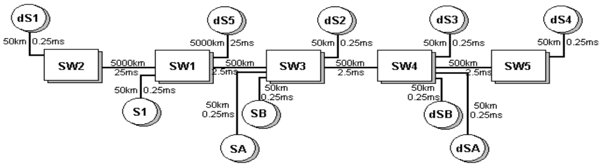

Figure 3. The Parking Lot Configuration

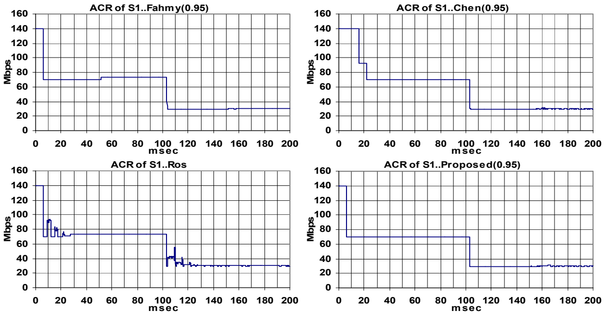

Figure 4. Results for P.L. Config. (high threshold)

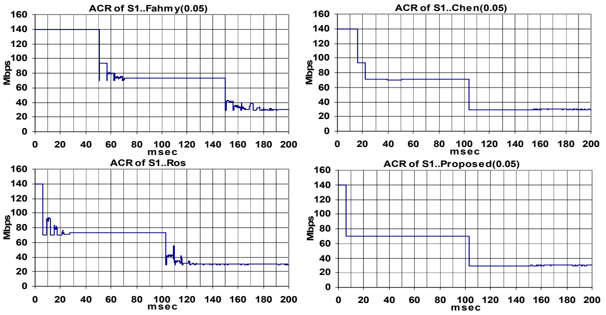

Figure 5. Results for P.L. Config. (low threshold)

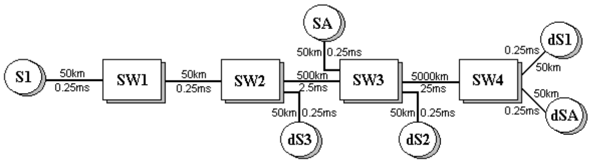

Figure 6. The Chain Configuration

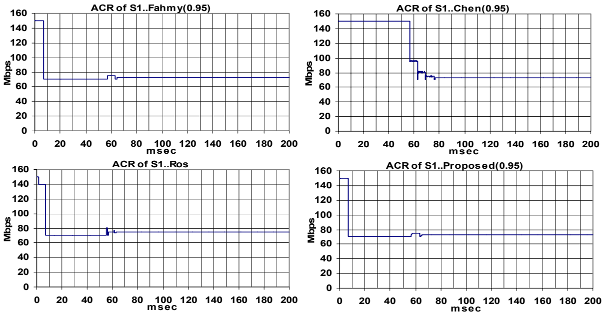

Figure 7. Results for Chain Config. (high threshold)

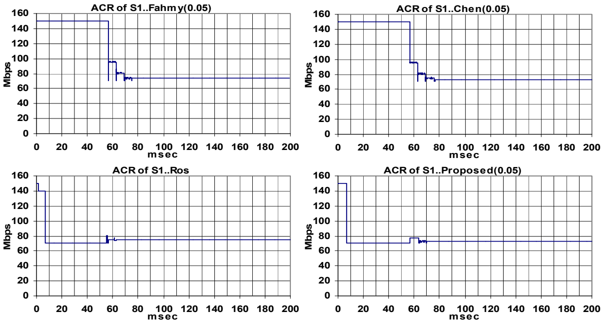

Figure 8. Results for Chain Config. (low threshold)

In case of low threshold, Fahmy and Chen algorithms yield worst performance. Fahmy algorithm waits for all branches to respond because the rate decrease is greater than 0.05. Chen is lacking the switch congestion situation thus waiting also for all branches to respond. The proposed algorithm keeps the same fast transient response as in case of high threshold.

5 Comparison of the Algorithms

In terms of complexity, our algorithm is more complex than both Fahmy's and Chen's, since it uses a hybrid approach to determine if an extra BRM cell will be sent or not. But its complexity is decreased by checking the local congestion state only one time per received FRM cell. The high complexity of Ros algorithm is due to storing M IDs and Bottleneck rates, and maintaining the BBM matrix.

All algorithms with high threshold offer reasonable fast transient response. However, Both Fahmy and Chen algorithms exhibit a slow transient response if the threshold is close to zero. While, our proposed algorithm provides adaptability to send an extra BRM cell with a probability p beside the periodic switch congestion check. In both cases of threshold value, the proposed algorithm exhibits fast response.

As for consolidation noise, all algorithms (except Ros), which are modified versions of the "wait-for-all" algorithm, eliminate the severe consolidation noise problem by waiting for feedback from all branches. Although, they all (except Ren's) may send extra BRM cells in cases of overload or at least rate decrease. This doesn't introduce noise, since the BRM cells only carry rate decrease information. Ros may suffer from little noise because of its sensitivity to any bottleneck change.

As for RM cell ratio, Fahmy and the proposed algorithm ensure the ratio is one over the long run. Chen algorithm has no limit for the ratio. Ros ratio converges quickly to one since it sends BRM cell at every N BRM cells received at most.

6 Conclusions

In this paper, we address the issue of feedback consolidation for point-to-multipoint connections in ABR service of ATM networks. Several consolidation algorithms have been proposed. The algorithms demonstrate a tradeoff between achieving fast transient response and reducing consolidation noise. Using simulation experiments, we have compared our proposed algorithm to the previously proposed algorithms. Results show that our proposed algorithm doesn't suffer from consolidation noise problem while exhibiting a fast transient response with accurate feedback in both cases of high and low thresholds (threshold-independent). The overhead of local congestion check is reduced in our proposed algorithm. It checks only once per new received FRM cell.

References

[1] Sonia Fahmy, Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Bobby Vandalore and Xiangrong Cai, "A Survey of Protocols and Open Issues in ATM Multipoint Communication", OSU Technical Report, Aug 21, 1997.

[2] L. Roberts. "Rate based algorithm for point to multipoint ABR service", ATM Forum 94-0772R1, November 1994.

[3] K-Y Siu and H-Y Tzeng, "Congestion Control for multicast service in ATM networks", Proceedings of the IEEE GLOBECOM, 1995.

[4] W. Ren, K-Y Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM", Proceedings of IEEE ATM Workshop, San Francisco, August 1996.

[5] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shivkumar Kalyanaraman, Sastri Kota, and Pradeep Samudra, "Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks", Infocom 98, San Francisco, March 1998.

[6] T. Jiang, E. W. Zegura, and M. Ammar, "Improved consolidation algorithms for point-to-multipoint ABR service", Proceedings of IEEE Workshop on ATM, May 1998.

[7] H-S. A. Chen, K. Nahrstedt, "Feedback Consolidation and Timeout Algorithms for Point-to-Multipoint ABR Service", Proceedings of IEEE International Conference on Communications (ICC'99), Canada, June, 1999.

[8] Jordi Ros, Wei K. Tsai and Mahadeven Iyer, "Bottleneck Branch Marking For Noise Consolidation In Multicast Networks", Proceedings of the IEEE International Conference on Networks (ICON'00), 2000.

[9] R. Jain, S. Kalyanaraman, S. Fahmy, R. Goyal, "The ERICA Switch Algorithm: A Complete Description", ATM Forum Contribution 96-1172, August 1996.