Improved Consolidation Algorithms for Point-to-Multipoint ABR Service *

Tianji Jiang **Ellen W. Zegura** **Mostafa Ammar**

Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{tjiang,ewz,ammar}@cc.gatech.edu

Abstract

The point-to-multipoint ABR service within ATM is important for many emerging data applications. The objective is for the source to operate at a rate representing its max-min fair share over the bottleneck link in the multicast tree over which the connection is established. Resource Management (RM) cells are used by receivers to convey back to the source an explicit rate indicating their bandwidth fair share on the bottleneck of the path leading to them. Switches in the multicast tree consolidate the information in backward RM cells received from downstream switches or receivers and forward this consolidated information to their upstream switch. Most proposed consolidation algorithms attempt to maximize either the speed of the return of feedback information to the source or the accuracy of the feedback. In this paper we first present a probabilistic consolidation algorithm that can be tuned using probability parameters to span the speed-accuracy spectrum. We then describe a technique that can be used to increase the accuracy of a broad class of switch algorithms. We demonstrate the performance of our proposed algorithms through a set of numerical examples.

1 Introduction

1.1 Problem Statement

The point-to-multipoint, or *multicast*, ABR service within ATM is important for many emerging data applications. While the details of providing such a service are the subject of on-going research, there is agreement on the basic structure of the service. The objective is for the source to operate at a rate representing its max-min fair share over the bottleneck link in the multicast tree over which the connection is established. Resource Management (RM) cells are used by receivers to convey back to the source an explicit rate (ER) indicating their bandwidth fair share on the bottleneck of the path leading to them. Switches in the multicast tree *consolidate* the information in backward RM (BRM) cells received from downstream switches or receivers and forward this consolidated information to their upstream switch[1]. The objective of the switch consolidation algorithms is to aggregate the information about downstream paths into a single explicit rate representing the minimum rate stated in the BRM cells they receive.

In the design of consolidation algorithms there are typically two conflicting objectives:

- *Speed:* A switch should convey any new information about changed rates towards the source quickly.
- *Accuracy:* The switch should insure the accuracy of aggregated feedback information by making sure it has heard from all downstream paths before it forwards aggregated information towards the source.

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. This results in a *long transient period* until the source is able to adjust its rate to accurately meet prevailing network conditions. 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* which can result in heavy oscillation of the cell rate used by the source.

Most consolidation algorithms attempt to maximize one side of the speed-accuracy tradeoff. In this paper we first present a probabilistic consolidation algorithm that can be tuned using probability parameters to span the speed-accuracy spectrum. We then describe a technique that can be used to reduce the consolidation noise of a broad class of switch algorithms. This reduction in consolidation noise allows for increased accuracy in the feedback information.

1.2 Related Work

A consolidation algorithm operating at a switch must address two issues: (1) when to generate a backward RM

*This work is supported in part by research grants from NSF, Intel and Hitachi Telecom (USA).

[1] The *downstream* direction from a switch is towards the receivers and the *upstream* direction is towards the source. Forward RM (FRM) cells travel downstream whereas backward RM (BRM) cells travel upstream.

(BRM) cell and (2) what information to convey in the BRM cell. Several consolidation algorithms have been proposed [1, 2, 3, 5]. All these algorithms demonstrate a tradeoff between achieving fast transient response and reducing consolidation noise. Some algorithms [1, 2] have a fairly fast transient response at the cost of higher consolidation noise, while other algorithms (algorithm 4 in [3]) may prolong the transient response period to reduce consolidation noise.

Robert [1] proposes that a switch at a branch point return a BRM cell to its upstream node whenever it receives an FRM cell. The explicit rate in the BRM cell is the minimum of the explicit rate in the last FRM cell and all explicit rates reported by downstream branches since the last BRM cell was generated. Robert's algorithm shows fast transient response. However, it may experience heavy consolidation noise due to inaccurate feedback and may lead to serious rate oscillation [2, 3].

Tzeng and Siu [2] present a slightly more conservative algorithm that remedies some of the inaccuracy problems of Robert's scheme. In the Tzeng and Siu 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. The explicit rate in the BRM cell is the same as in Robert's algorithm, i.e., the minimum of the explicit rate in the last FRM cell and all explicit rates reported to the switch since the last BRM was generated. Using this algorithm, more accurate feedback can be achieved and consolidation noise can be reduced somewhat. However, the response to local switch overload conditions is slower than in Robert's scheme, since the switch must wait for a BRM cell to arrive from downstream before reporting on local overload [3].

Ren et al. [3] argue that the algorithms described thus far may introduce heavy RM cell processing overhead on switches. Therefore, they propose an algorithm in which switches do not initiate generation of BRM cells (as in [1, 2]), but instead simply forward selected BRM cells received from downstream, after modifying the contents. For example, Algorithm 3 proposed by Ren et al. 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, and the problem of consolidation noise still exists.

A natural extension to the Ren et al. algorithm [3] is for a switch to return a BRM cell only after receiving feedback information from *all* its downstream branches since the last BRM cell was sent (algorithm 4 in [3]). 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 can easily eliminate the consolidation noise, but the resulting slow transient response may not be satisfactory.

In the sequel, we compare our proposed algorithms to the algorithms of Robert [1], Tzeng and Siu [2] and Ren et al. Algorithm 3 [3]. We briefly mention one relatively new set of algorithms, though they are not included in our simulation study. Fahmy et al. [5] argue that *excessive* overload information should be reported back to the source as soon as possible without the need to wait for feedback from all other branches. This motivates the proposal of three feedback consolidation algorithms aimed at avoiding slow transient response. Their algorithms improve the consolidation process by using (possibly) extra BRM cells to pass overload information back to source more quickly whenever an overload situation is detected.

2 The Probabilistic Aggregation Scheme

A close examination of existing consolidation algorithms reveals that Robert's [1], Tzeng and Siu's [2] and algorithm 3 by Ren et al. [3] are targeted towards generating fast feedback to the source and inherently cannot reduce consolidation noise. The only algorithm that explicitly avoids consolidation noise is the fourth "wait-for-all" algorithm proposed by Ren [3]: the switch waits for feedback from *all* immediate downstream nodes before a BRM cell is sent upstream. However, this scheme results in a slow adjustment of source rate.

Our proposed *Probabilistic Aggregation* technique preserves some of the advantages of the wait-for-all algorithm and at the same time employs mechanisms to improve its slow transient response. We modify the wait-for-all algorithm in the following way:

> For each FRM cell extra BRM cells *may* be sent upstream during the consolidation procedure (i.e., before feedback from all downstream paths is received) so that any detected change in conditions can be passed back to the source more quickly.

This scheme works as follows. The switch maintains a counter for each downstream branch, initialized to zero. When a BRM cell arrives, the switch increments the corresponding counter by one and recalculates the minimum explicit cell rate reported so far. The switch then checks to see if all of the counters have value greater than zero, thus indicating that BRM cells have been received from all downstream branches. If this is the case then a BRM cell is sent upstream with the minimum explicit rate reported, and all counters are reduced by one. The scheme as described thus far is exactly the same as wait-for-all. The difference arises if at least one counter has value zero. In this case, with probability p the switch forwards a BRM with the minimum ER reported. That is, the switch has some chance of sending a BRM cell while waiting for feedback from all downstream branches. The value of p can be adjusted to navigate the speed-accuracy tradeoff. When $p = 0$ then the scheme becomes equivalent to the *wait-for-all* algorithm and the scheme achieves maximum accuracy. When $p = 1$ then the scheme achieves maximum speed and suffers high consolidation noise.

The algorithm is shown in Figure 1. The parameter p determines the probability that extra BRM cells are generated. At a switch, the following data structures are kept for each multicast virtual connection:

- Two registers, the current minimum explicit rate (MER) and the congestion indication bit (MCI). MER is initialized to PCR; MCI is initialized to false.
- M counters, where M is the number of (immediate) downstream branches associated with this switch in multicast tree. Each counter is initialized to zero.

```
Upon the receipt of forward RM(ER,CI) cell:
     Multicast this RM cell to all participating
               branches;

Upon the receipt of backward RM(ER,CI) cell:
     Increment by one the counter corresponding to
            the branch on which BRM was received;
     if (RM-ER < MER) then DecER=True;
     If (all M counters > zero) then:
            ER=min(ER,MER); CI=or(CI, MCI);
            Send this RM(ER,CI) cell back
                 to the source;
            Decrease M counters by one;
            Reset MER=PCR, MCI=False;
            DecER=False;
     else {
          /* A BRM cell is not received from some
                    downstream branch */
          if (RandomValue<p) AND (DecER) then
             ER=MER=min(ER, MER);
             CI=MCI=or(CI, MCI);
             Send this RM(ER,CI)
                  cell back to source
             DecER=False;
          else
             MER=min(ER, MER);
             MCI=or(CI, MCI);
             Discard this RM cell;
     }
```

Figure 1: Probabilistic Aggregation Scheme

- A Decrement flag (DecER), used to indicate whether a more restrictive explicit rate has been received since the last BRM cell was sent upstream. An extra BRM cell may be generated only if DecER is true. DecER is initialized to false.

In Section 4 we will show the performance of this probabilistic aggregation algorithm as compared to other proposed algorithms. For the purposes of the experiments we propose using a value of $p = 1/M$ where M is the number of downstream tree branches for the multicast connection at that switch. Thus, on average one extra BRM cell will be sent during consolidation, regardless of the number of downstream branches.

3 Memory-Enhanced Consolidation

In all of the consolidation algorithms described thus far, including our Probabilistic Aggregation scheme, the switch consolidates the feedback information reported from the downstream branches and records it in an MER register. This information is included in the next BRM cell sent by the switch. However, all of these algorithms have a common defect: *at a switch, all stored information is cleared whenever a BRM cell is sent upstream.* Specifically, the current minimal explicit rate is reset to the Peak Cell Rate (PCR). In fact, the recently-received ER

```
Upon the receipt of forward RM(ER, CI) cell:
     Multicast this RM cell to all
               participating branches;
     Calculate MER=min(ER_i), i = 1,...,M, and
     Return a BRM(ER=MER,CI=MCI) cell to
               source;
     Reset MCI=0;

Upon the receipt of backward RM(ER, CI) cell:
     Record RM-ER into its corresponding
            ER-register, i.e, overwrite
            previous value;
     MCI = or(MCI, RM-CI);
     Discard this RM cell
```

Figure 2: Memory-Enhanced Algorithm

information for a branch may be a good indication for the future ER from that particular branch. By resetting the cell rate to PCR at the end of an iteration, these algorithms remove all historical information and re-initialize the default value of ER to PCR.

We propose an enhancement to the consolidation procedure that can be applied to many proposed algorithms. In the enhancement, switches remember the previous explicit rates reported on their downstream branches. Until a new explicit rate is reported, the old value is used by the consolidation procedure. This enhancement is straightforward to implement and has the effect of improving the accuracy of a consolidation algorithm without any reduction in its speed. We will show the effect of this enhancement on the performance of some consolidation algorithms in Section 4.

Fig.2 shows how our proposed memory-enhancement can be added to Robert's algorithm. A similar addition can be made to any of the other consolidation algorithms. For each switch, the following data structures are kept for each multicast virtual connection:

- Two registers, the current minimum explicit rate (MER) and the congestion indication bit (MCI). MER is initialized to PCR; MCI is initialized to false.

- M ER-registers, where M is the number of (immediate) downstream nodes associated with this switch in this multicast tree. Each ER-register stores the most recent value of ER reported from the downstream branch. Each ER-register is initialized to the Peak Cell Rate (PCR).

We find that the memory-enhancement allows the algorithms with fast transient response time to approach the accuracy of the wait-for-all algorithm. That is, the use of past history means that a memory-enhanced algorithm will always include information from all branches, albeit sometimes stale information.

We also note that the memory-enhancement provides a measure of robustness for slow or non-responsive downstream branches. The basic wait-for-all algorithm will be significantly delayed while waiting for a slow branch to respond. The memory-enhancement added to, for example,

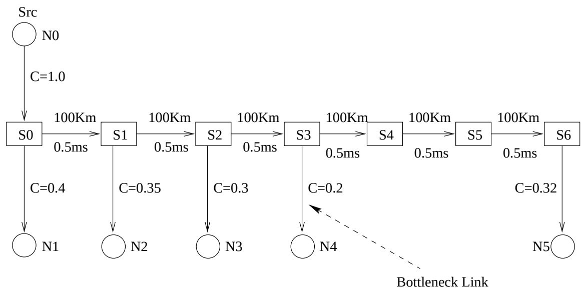


Figure 3: Chain Configuration

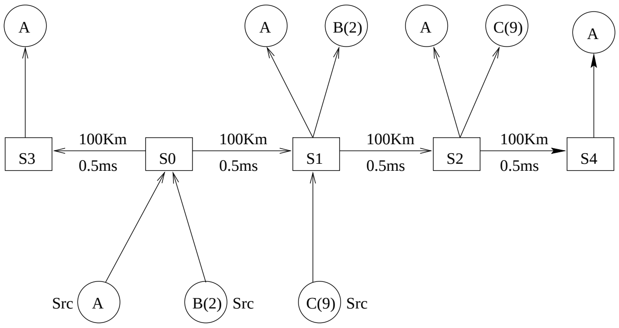


Figure 6: Parking Lot Configuration

Robert's scheme can approach the accuracy of wait-for-all without the delay due to a slow branch.

4 Performance Results

We next present a selected set of results comparing the performance of our two new algorithms to algorithms from the literature. In the results that follow, "Robert" denotes Robert's algorithm [1]; "TS" denotes the Tzeng and Siu algorithm [2]; and "RSS" denotes the third Ren, Siu and Suzuki algorithm [3]. For the Robert and Tzeng-Siu algorithms, we assume that the explicit rate in all FRM cells is the peak cell rate (PCR). Thus, the contents of the BRM cells generated is the minimum of the PCR and all explicit rates reported by downstream branches since the last time a BRM cell was generated.

Unless otherwise noted, the following assumptions hold for all simulations:

- The connections are uni-directional; the only traffic from receivers to sources is BRM cells. In the configurations, the arrows on the links are in the direction of the data traffic.
- Available bandwidth capacity on each link is normalized to the range $[0.0, 1.0]$. Unless otherwise stated, the available capacity is 1.0.
- The propagation delay on the link from a source or receiver to a switch is assumed to be 0. Propagation delays between switches are explicitly given.
- The sources are persistent, i.e., there is always data to send, therefore the sources always attempt to send at the highest rate allowed.
- The initial source rate is .05, and the peak source rate is 0.9.
- The source rate increase factor (RIF) (RIF controls the amount by which the cell transmission rate may increase upon receipt of an RM cell at source) is set to 1.0 to allow immediate use of the explicit rate in an RM cell.

4.1 Chain Configuration

The first configuration is shown in Figure 3. S_0 to S_6 denote six ATM switches. N_0 is the source of the multicast session with five receivers, N_1 to N_5. The bottleneck link is the one between switch S_3 and leaf N_4, with available capacity 0.2.

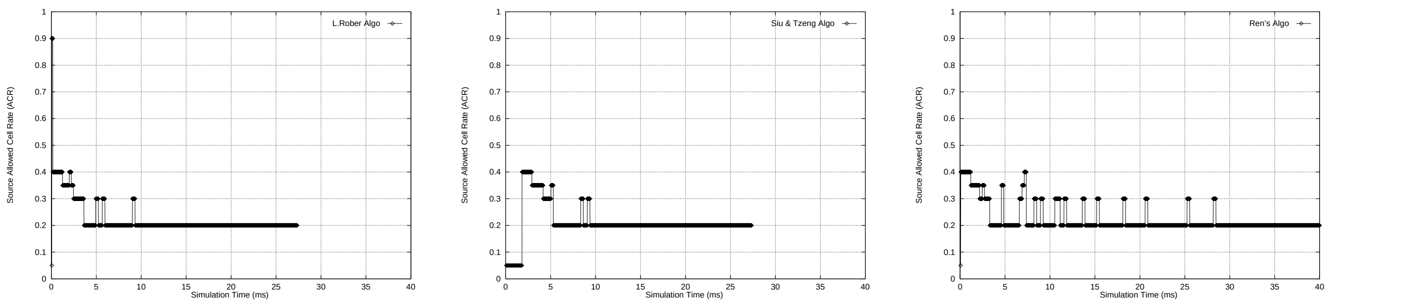
Based on the topology, we can see that the minimum time for the source to achieve the bottleneck rate (assuming BRMs are triggered by FRMs) is 3 ms, since this is the round-trip time from the source to the receiver N_4. The wait-for-all algorithm will converge in 6 ms, since this is the round-trip time from the source to the farthest receiver.

Figures 4 and 5 illustrate the performance of the three algorithms from the literature and the two new algorithms, respectively. The probabilistic algorithm uses a probability of 1/M, where M is the number of downstream branches. The memory-enhanced algorithm is used with Robert's scheme to determine the generation of the BRM cells. These figures plot the source rate over time, with source adjustment dictated by the RM cells returned.

The salient features of these plots are as follows. The Robert and TS algorithms both initially reach the target rate relatively quickly, however both proceed to oscillate between 0.2 and 0.3 before converging after 9 ms. The initial increase to the peak cell rate (0.9) experienced by Robert's algorithm is caused by the first FRM cell that triggers a return BRM cell before any feedback has been received from downstream. Recall that the TS algorithm will not generate a BRM cell until some downstream feedback has been received, and thus does not experience the same initial peak value. The RSS algorithm exhibits oscillation that lasts considerably longer, until about 28 ms. The two new algorithms progressively approach the target rate without oscillating and achieve the target after the minimum of 3 ms.

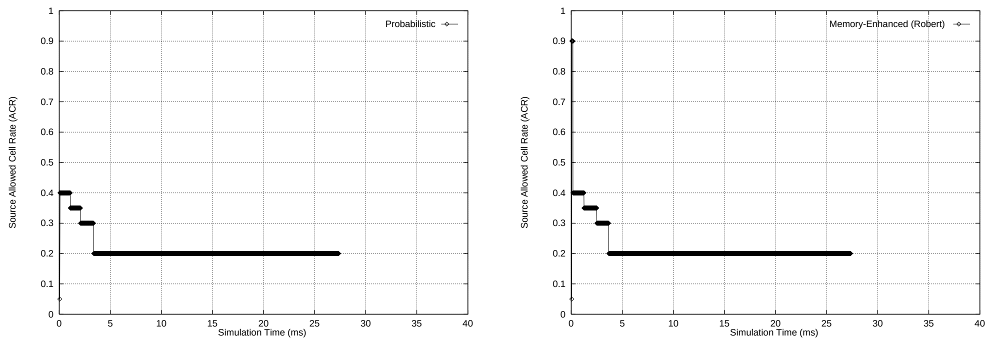
4.2 Parking Lot Configuration

The configuration for the second simulation is shown in Figure 6. S_0 to S_4 denote five ATM switches. This configuration has one active multicast session, with source and four receivers marked A. The configuration has 11 unicast sessions; two of these sessions go between the nodes labeled B(2) and nine of these sessions go between the nodes labeled C(9). The bottleneck link for the multicast session is the link between switches S1 and S2. The multicast session shares this link with the nine unicast connections labeled C, thus the bottleneck rate is 0.1. The minimum time for the multicast source to achieve the bottleneck rate is 2 ms; the wait-for-all algorithm will converge in 3 ms.



(a) Robert (b) TS (c) RSS

Figure 4: Conventional Algorithms: Chain Configuration



(a) Probabilistic (b) Memory-Enhanced (Robert)

Figure 5: New Algorithms: Chain Configuration

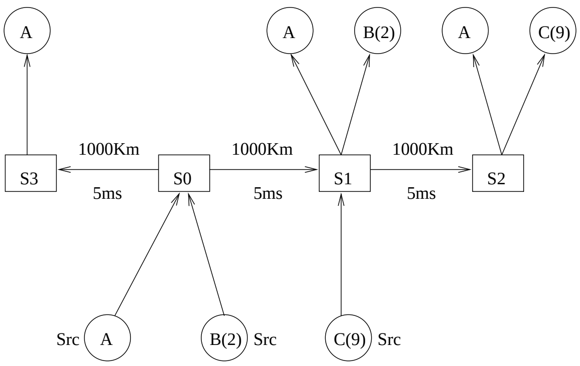


Figure 9: Modified Parking Lot Configuration

From simulation results in Figure 7 to Figure 8, we can see that the Probabilistic aggregation and memory-enhanced algorithms almost have the same transient time, and show better performance than algorithms from the literature. In this case, the original Robert's algorithm produces relatively little oscillation; the oscillation that does exist is removed by the memory-enhancement.

4.3 Modified Parking Lot Configuration

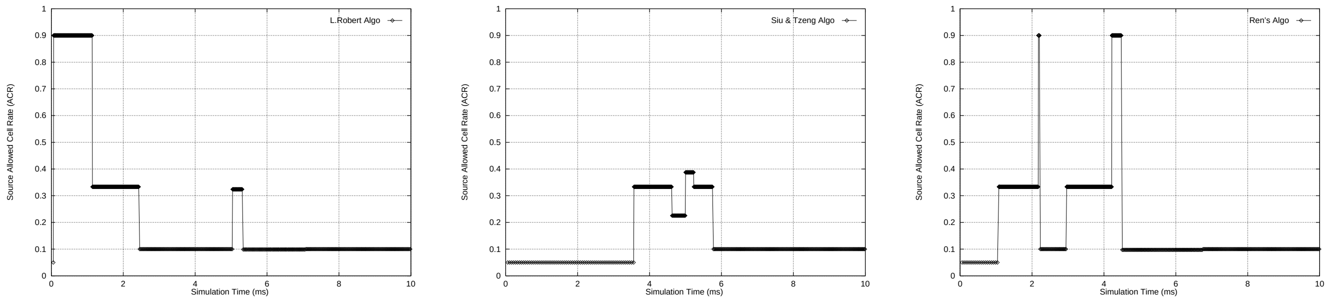
We turn our attention to the memory-enhancement and the use of this algorithm with a variety of schemes to determine the generation of the BRM cells. For this next experiment, we use the modified parking lot configuration shown in Figure 9. The set of sessions is the same as for the original parking lot (i.e., one multicast session labeled A, two unicast sessions labeled B and nine unicast sessions labeled C). The bottleneck link for the multicast session is the link between switches S1 and S2, with a target bandwidth of 0.1.

Based on the topology, the fastest convergence time is 20 ms; because the bottleneck link is the farthest from the multicast source, the wait-for-all algorithm will also require 20 ms to converge.

Figures 10 and 11 show the rate at the source for session A over time, with and without the memory-enhanced addition to the basic algorithms. Without the memory-enhancement, the convergence times vary from 20 ms (Robert) to more than 30 ms (TS and RSS). Both the TS and RSS schemes also exhibit considerable oscillation between rates of 0.1 and 0.33. (Note that 0.33 is the fair share for session A on the link from S0 to S1.) For this configuration, the memory-enhancement has little effect on Robert's algorithm, however it reduces the convergence time and oscillation considerably for the TS and RSS algorithms. Oscillation is nearly eliminated, and convergence occurs in almost the minimum time of 20 ms.

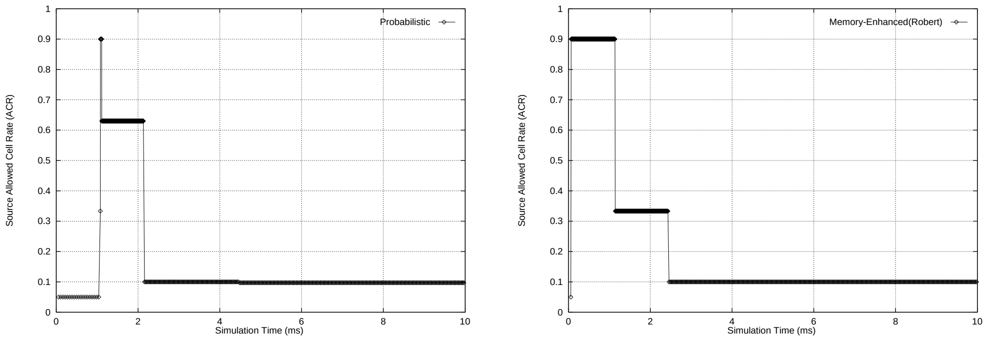
4.4 Modified Parking Lot with Bursty Traffic

Finally, we examine the effect of bursty traffic on the source rate. We keep the modified parking lot configu-



(a) Robert (b) TS (c) RSS

Figure 7: Conventional Algorithms: Parking Lot Configuration



(a) Probabilistic (b) Memory-Enhanced (Robert)

Figure 8: New Algorithms: Parking Lot Configuration

ration in Figure 9, however we alter the unicast sources to be bursty. Specifically, the two sources labeled B are only active from time 0 ms to 43.5 ms. The nine sources labeled C are only active from time 0 ms to 30 ms. Thus, the target rate for multicast source A is 0.1 until time 30 ms, 0.33 from 30 ms to 43.5 ms, and 0.9 (the peak cell rate) after 43.5 ms.

Figures 12 and 13 show the rate at the source for session A over time, with and without the memory-enhancement. We see that in all cases, the memory-enhancement virtually eliminates the oscillation and speeds convergence.

5 Conclusions

We have examined the design space for consolidation algorithms and determined that most algorithms are either (1) quick to initially reach the target rate, yet oscillate considerably thereafter or (2) slow to reach the target rate, yet exhibit little oscillation. We developed a probabilistic consolidation algorithm that can be tuned using a probability parameter to span the speed-accuracy spectrum. We also developed a technique to reduce a common problem in fast algorithms, namely oscillation after initially reaching the target rate.

Through a set of simulations, we compared the performance of our new algorithms to a set of schemes from the literature. A limited set of those results were reported here. We consistently found that our probabilistic algorithm reached the target rate at least as fast (and often faster) than any of the other schemes with which we compared. This gain comes at the cost of slightly higher implementation complexity, primarily random number generation. Our straightforward memory-based enhancement technique was able to nearly eliminate oscillation and reduce convergence time when used in conjunction with the other schemes.

References

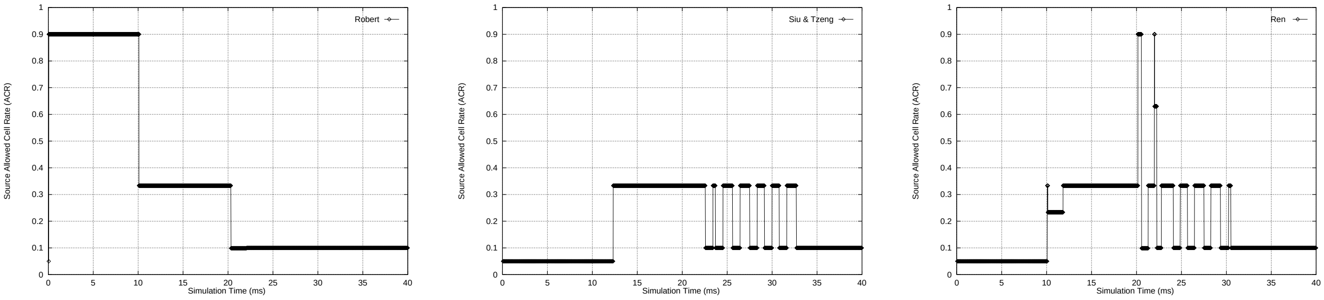
[1] L. Roberts. Rate Based Algorithm for Point to Multipoint ABR Service. *ATM Forum Contribution 94-0772*, 1994.

[2] H. Y. Tzeng and K. Y. Siu. On Max-min Fair Congestion Control for Multicast ABR Service in ATM. *IEEE Journal on Selected Areas in Communications*, Vol.15, No.3, April 1997.

[3] W. Ren, K. Siu and H. Suzuki. On the Performance of Congestion Control Algorithm for Multicast ABR Service in ATM. *IEEE ATM '96 Workshop*, San Francisco, August, 1996.

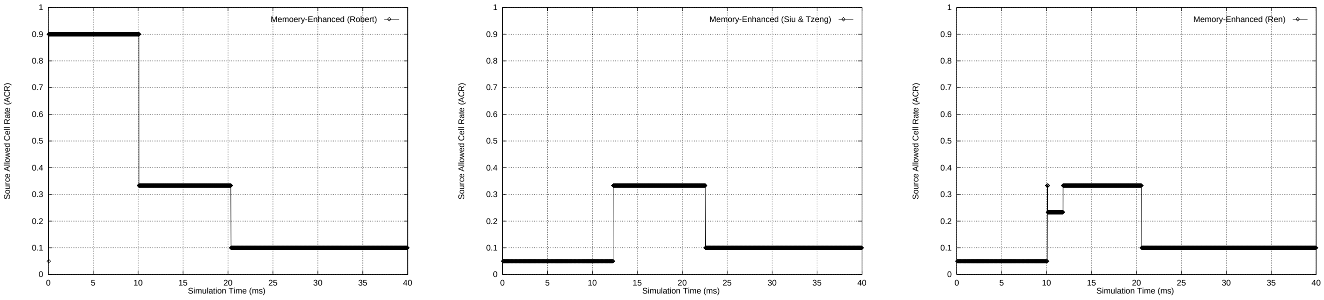
[4] ATM Forum. ATM Forum Traffic Management Specification 4.0. 1996

[5] S. Fahmy, R. Jain and et al. Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks. *IEEE Infocom'98*, San Francisco, March, 1998.



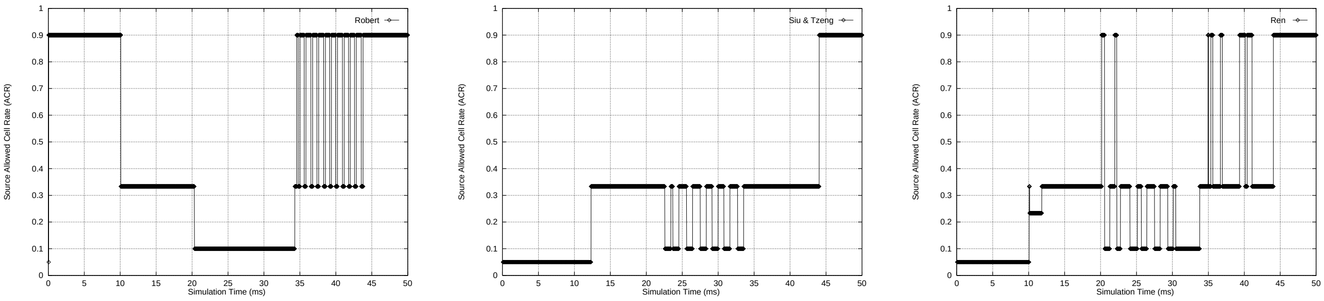
(a) Robert (b) TS (c) RSS

Figure 10: Conventional Algorithms: Modified Parking Lot Configuration



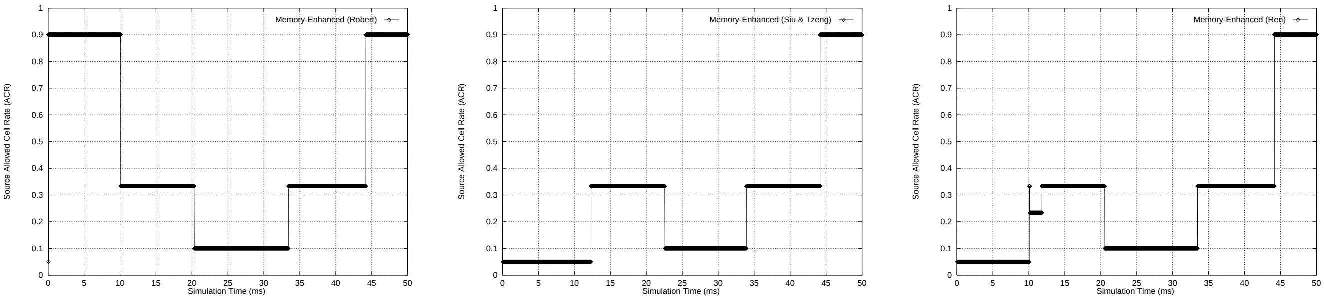
(a) Memory-enhanced (Robert) (b) Memory-enhanced (TS) (c) Memory-enhanced (RSS)

Figure 11: Memory-Enhanced Algorithms: Modified Parking Lot Configuration



(a) Robert (b) TS (c) RSS

Figure 12: Conventional Algorithms: Bursty Traffic



(a) Memory-enhanced (Robert) (b) Memory-enhanced (TS) (c) Memory-enhanced (RSS)

Figure 13: Memory-Enhanced Algorithms: Bursty Traffic