Abstract

ABR traffic management for point-to-multipoint connections controls the source to the minimum rate that can be supported by the switches on the paths from the source to all of the leaves in the multicast tree. A number of algorithms have been proposed for extending ABR congestion control algorithms in ATM networks to perform feedback consolidation at the branch points. This paper discusses various design options and compares the performance of the developed algorithms under a variety of configurations. The effect of the parameter, threshold value, is also studied wherever applicable. Results indicate that there is a tradeoff involved in the accuracy, speed, overhead and consolidation noise of the algorithms.

Keywords: Asynchronous Transfer Mode, Available Bit Rate Service, Multicast, Branchpoint Algorithms, Congestion Control.

1. Introduction

The Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks is suitable for non real-time applications like data communications, which do not have stringent delay or loss requirements. The key attractive features of ABR are that it gives sources low cell loss guarantees, minimizes queuing delay, provides possibly non-zero minimum rate guarantees, utilizes buffers and bandwidth efficiently, and gives the contending sources fair share of the available resources. 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. There is no specified contract between the network and client that describes the traffic behavior and expected Quality of Service (QoS). The ABR service dynamically assigns bandwidth to the contending sources. So, ABR requires a closed loop congestion control scheme, dynamically regulating the cell transmission.
rate of each source according to congestion status.

Multicast in ATM is useful because it allows the construction of truly distributed applications and provides important performance optimization over unicast transmission. It has a wide range of applications in audio and video conferencing, server and database synchronization, LAN emulation, distributed computing applications like Distributed Interactive Simulation (DIS), distribution of software, etc.

The ABR service for data traffic in ATM networks periodically indicates to sources the rate at which they should be transmitting. The switches monitor their load and compute the available bandwidth, dividing it fairly among active flows. The feedback from the switches to the sources is sent in Resource Management (RM) cells which are generated by the sources and turned around by the destinations. The RM cells contain the source Current Cell Rate (CCR), in addition to fields that can be used by the switches to provide feedback to the sources. These fields are: Explicit Rate (ER), the Congestion Indication (CI) flag, and No Increase (NI) flag. The ER field indicates the rate that the network can support at this particular instant. Initially, the ER field is set to a value no greater than the Peak Cell Rate (PCR), and the CI and NI flags are clear. Each switch on the path reduces the ER field to the maximum rate it can support, and sets CI or NI if necessary [1].

Although point-to-multipoint connection is an extension to the point-to-point connection, the traffic management problems it faces are not exactly the same as point-to-point connections do. In particular, the consolidation of the feedback information from the different leaves of the tree is necessary for point-to-multipoint connections. This is due to the feedback implosion problem, i.e., feedback information provided to the sender should not increase proportionally to the number of leaves in the connection. The next section presents the working of the previously proposed feedback consolidation algorithms that have been implemented in order to compare their performance. The third section lists the details of the simulation experiment like parameter settings and then presents the results and analysis for two different network configurations. The final section concludes the paper.

2. Congestion Control Algorithms

In the algorithms presented, a rate allocation scheme, ERICA (Explicit Rate Indication Congestion Avoidance) is employed before sending the BRM (Backward Resource Management) cells in order to ensure that the most recent feedback information is sent. ERICA operates at each output port of a switch and periodically monitors the load on each link to allocate the bandwidth fairly among the contending sources [2].

2.1 Algorithm 1

In order to achieve a fast transient response, it was suggested that there is no need to wait for feedback from all the branches when a severe overload situation has been detected. The overload should be immediately indicated to the source. In cases of underload indication from a branch, it is better to wait for feedback from all branches, as other branches may be overloaded. Overload is detected here when the feedback to be indicated is much less than the last feedback returned by the branch point. This condition is tested using a multiplicative factor, threshold. The threshold value can range from zero to one [3].

A register maintains the last explicit rate value returned by the branch point. It is stored per multipoint VC and compared to the value of MER (Minimum Explicit Rate).
Another register is used to control the RM cell ratio. It 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 this register is greater than zero, this particular feedback is ignored and the register is decremented. In cases when the branch point is itself a switch and queuing point, this method accounts for the potential overhead at the switch by invoking the rate calculation algorithm whenever a BRM is received, and not just when a BRM is being sent.

2.2 Algorithm 2

The next scheme is a probabilistic consolidation algorithm that can be tuned using probability parameters to span the speed-accuracy spectrum [4]. 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. The value of probability that an overload feedback is sent, can be adjusted to navigate the speed-accuracy tradeoff. An extra BRM cell may be generated only if a more restrictive explicit rate has been received since the last BRM cell was sent upstream. The feedback implosion problem may appear because extra BRM cells are sent and no handling is presented to balance the FRM to BRM ratio [4].

2.3 Algorithm 3

The next algorithm that has been implemented is the one proposed in [5] which attempts to alleviate the threshold problem by providing more flexibility to span the speed overhead spectrum. This scheme sends an extra BRM cell with a probability, which is a function of the current collected MER and the last returned feedback. The probability to send an overload feedback is a linear function between two ends. An extra BRM cell is sent if an overload condition is detected.

2.4 Algorithm 4

In the previous schemes proposed, the only way to increase MER is to wait for it to be reset. The Bottleneck Branch Marking (BBM) algorithm [6] keeps track of the M most bottlenecked branches. The switch stores a matrix with M entries including both the identifier of the branch (ID) and the last available rate received at the branching node. 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. Now the switch can identify that the most bottlenecked branch has increased its available rate. The trade-off between performance and complexity is reflected in the parameter M. A BRM cell may be sent when a new bottleneck appears or when feedback has been received from all branches and no extra BRM cell has been sent. This method has zero response delay, noise stability, and small probability of noise.

2.5 Algorithm 5

The last algorithm simulated is the probabilistic algorithm given in [7] in which there are two ways for the switch to trigger sending a BRM cell. Firstly, it returns a BRM cell if feedback is received from all branches. Secondly, an extra BRM cell is passed to the source if either the switch itself is overloaded or feedback indicating overload, is received from a branch. To alleviate the threshold sensitivity problem, the overload is checked using a probability function which is a function of the current collected MER and the last returned feedback. Congestion check at the branch point is only performed when receiving the
3. Performance Analysis

This section provides a limited set of results, obtained using simulation. These results compare the performance of the five algorithms given above, in terms of two performance metrics:

- **Allowed Cell Rate (ACR):** The Allowed Cell Rate is used to show a source transmission rate as a function of time. It is expressed in Mbps, rather than cells per second, to facilitate direct comparison with link capacity used in each scenario.
- **Queue Length:** The queue length shows queue occupancy of the switch output buffer as a function of time. It is expressed in number of cells.

The algorithms were simulated for two different threshold values (0.05 and 0.95) for each configuration.

3.1 Parameter Settings

Throughout our experiments, the following parameter values are used:

- All links have a bandwidth of 155.52 Mbps.
- All point-to-multipoint traffic flows from the root to the leaves of the tree. No traffic flows from the leaves to the root, except for RM cells. The same applies for point-to-point connections.
- All sources are deterministic, i.e., their start/stop times and their transmission rates are known. Variable Bit Rate (VBR) sources are on/off sources, where the on and off times are 20 ms.
- The source parameter Rate Increase Factor (RIF) is set to one, to allow immediate use of the full explicit rate indicated in the returning RM cells at the source.
- Initial Cell Rate (ICR) is also set to a high value (almost peak cell rate), except when indicated. These factors are set to such high values to simulate a worst case load situation.
- The source parameter Transient Buffer Exposure (TBE) is set to large values to prevent rate decreases due to the triggering of the source open-loop congestion control mechanism. This is done to isolate the rate reductions due to the switch congestion control from the rate reductions due to TBE.
- The switch target utilization parameter is set at 90%. The switch measurement interval is set to the minimum of the time to receive 100 cells and 1 ms.
- The explicit rate calculation algorithm used in the simulations is ERICA.
- The parameter M in algorithm 4 [6] is set to the minimum of 2 and number of branches of the multicast connection at the switch.

3.2 Simulation Results and Analysis

3.2.1 Chain Configuration

The first network configuration considered is the chain configuration illustrated in Figure 1. This configuration has four switches (SW1 to SW4). S1, SA are the sources and dS1, dS2, dS3, dSA are the destinations. It 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 in this configuration as the link connecting SW3 to SW4 is the bottleneck link. Figures 2 and 3 illustrate the performance of the algorithms for a high threshold. Figure 4 shows the performance of algorithm 1 for low threshold.

Figure 1: Chain Configuration

Figure 2: ACR of S1 (high threshold)
Algorithm 1 and 5 yield optimal performance in this case, since the rate of the source S1 immediately drops to its optimal value, as soon as the overload is detected. The bottleneck rate is \((0.9 \times 155.52)/2 \approx 70\) Mbps and the rate decrease factor is 0.5. 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 time from S1 to dS2). The maximum queue size and the queue length at the end of 200 msec is constrained to small values by these algorithms. This is because they check for the local congestion at the switch before sending feedback. Algorithm 3 suffers from slow transient response. The rate only drops after 56.5 ms, and by that time, large queues have built up at the switches. This is because SW3 must wait for a BRM cell from SW4.

The first rate decrease indicated in ACR (Allowed Cell Rate) graphs of algorithm 4 is not detected by other algorithms since the rate decrease ratio is 1. It shows fast response and low consolidation noise. This drop appears because ICR=150 (not 140) Mbps in this configuration. Algorithm 2 exhibits a slow response with little consolidation noise. It has large queues as it does not take the local congestion state of the switch into account.

![ACR of S1 (Algo 1)](a)

![Queue Length (Algo 1)](b)

Figure 4: Algorithm 1(low threshold)

Performance of the above algorithm degrades to the wait-for-all algorithm (slow transient response and huge initial queues) 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. The algorithm waits for all branches to respond because the rate decrease is greater than 0.05, which is the threshold value specified.

### 3.2.2 Jumping Bottleneck Configuration

The second configuration used is the jumping bottleneck configuration shown in Figure 5. The configuration has one point-to-multipoint connection (S1 to dS1, dS2, …, and dS10). In order to simulate bottlenecks moving from branch to branch, VBR traffic is added at each branch of SW2. The transmission rates for VBR sources are 5, 35, 65, 90, and 40 Mbps and their initial transmission times are 0, 8, 16, 24, and 32 ms. Length of all links from the end systems to the switches is zero. The ICR is 20 Mbps, which is a small rate thus we ignore the queue length graphs.
Figure 5: Jumping Bottleneck Configuration

The sequence of events of the configuration in the transient period is shown in Table 1.

Table 1: Events of Jumping Bottleneck Configuration

<table>
<thead>
<tr>
<th>Event No.</th>
<th>Time (msec)</th>
<th>Event</th>
<th>New Bottleneck Switch and Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>V1 becomes active</td>
<td>SW3, 135.5</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>V2 becomes active</td>
<td>SW5, 108.5</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>V3 becomes active</td>
<td>SW7, 81.5</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>V1 becomes inactive</td>
<td>SW7, 81.5</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>V4 becomes active</td>
<td>SW9, 59</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>V2 becomes inactive</td>
<td>SW9, 59</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>V5 becomes active</td>
<td>SW9, 59</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>V3 becomes inactive</td>
<td>SW9, 59</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>V1 becomes active</td>
<td>SW9, 104</td>
</tr>
<tr>
<td>10</td>
<td>44</td>
<td>V4 becomes inactive</td>
<td>SW11, 104</td>
</tr>
<tr>
<td>11</td>
<td>48</td>
<td>V2 becomes active</td>
<td>SW11, 104</td>
</tr>
</tbody>
</table>
The results shown in Figure 6 are obtained for a threshold value of 0.95. The source will not receive any response till 15 ms, which is the round-trip time from S1 to any of the bottleneck switches. The optimal rates at the source are lagging by only time from the bottleneck switch to the source. The algorithm 2 in this situation shows a very slow response and the source rate starts rising only after around 25 ms. There are a lot of rate oscillations which lead to performance degradation. This is because of the reduced chance to send a BRM cell which is calculated as a linear function. The algorithm 3 detects the first peak with probability of $(1-0.968)/(1-0.95) \approx 0.64$, but the source must wait for the round-trip longest destination (25 ms) until it receives feedback. This is due to the delay of switch congestion situation until taking the decision of sending BRM cell. This algorithm 4 has an optimal performance since SW2 has the values of the two most bottleneck switches and their available rates in hand all the time. Algorithm 5 detects the first peak with probability of $(1-0.968)/(1-0.95) \approx 0.64$. Its performance is near the optimal. Figure 7 depicts the performance of all the five algorithms for a low threshold value.

The performance of algorithm 1 again degrades to the wait-for-all performance. SW2 waits 20 ms to collect information from
all branches in absence of overload indication. It does not detect the first peak since the rate decrease ratio is 0.968 which is more than threshold of 0.05. Thus, it is very threshold sensitive. For algorithm 3, in the absence of overload indication and local congestion check, SW2 waits 20 ms to collect information from all branches, resulting in very slow response. Algorithm 2 exhibits low transient response and considerable consolidation noise as the probability of sending overload feedback is between 0 and 1.

Figure 7: ACR of S1 (low threshold)
The performance is near optimal in algorithm 4. Note that the graph obtained is the same as that for a high threshold value. Thus, algorithm 4 is threshold independent. The source rate starts increasing as early as 10 msec when the feedback information becomes available in the network. In the case of algorithm 5, the transient response is faster than of algorithm 1 and algorithm 3. Its performance is the nearest to algorithm 4. By the periodical local congestion check beside the chance to send BRM cell if rate decrease is above the threshold, this algorithm tracks the optimal solution from the second peak at around 17 ms. Note that the probability to send BRM cell in the first peak is $(1-0.97)/(1-0.95) = 0.03$, which is very low.

3.3 Comparison of Algorithms

This section summarizes the performance comparison of the algorithms. The comparison is based on complexity, transient response, consolidation noise and RM ratio of algorithms. Table 2 compares the different algorithms.

Table 2: Comparison between the algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Complexity</td>
<td>&gt;Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>&gt;Medium</td>
</tr>
<tr>
<td>Transient Response (low threshold)</td>
<td>Slow</td>
<td>Slow</td>
<td>Slow</td>
<td>Very fast</td>
<td>Fast for overload</td>
</tr>
<tr>
<td>Transient Response (high threshold)</td>
<td>Very Fast for overload</td>
<td>Fast for overload</td>
<td>Fast for overload</td>
<td>Very fast</td>
<td>Very fast for overload</td>
</tr>
<tr>
<td>Consolidation Noise</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>BRM:FRM at root</td>
<td>Lim =1</td>
<td>May be &gt;1</td>
<td>May be &gt;1</td>
<td>Lim =1</td>
<td>Lim =1</td>
</tr>
</tbody>
</table>

In terms of complexity, algorithm 1 has more than average complexity. Algorithm 5 is more complex than both algorithm 1, 2 and 3, 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 algorithm 4 is due to storing M IDs and bottleneck rates, and maintaining the BBM matrix.

When an overload is detected, algorithm 1 has a fast response, if threshold is set to a high value. Otherwise, its performance deteriorates to the wait-for-all algorithm. All algorithms with high thresholds offer reasonable fast response. The response of algorithm 3 is fast for overload when threshold is high. However, it exhibits a slow transient response if the threshold is close to zero. Algorithm 4 is threshold independent. While, algorithm 5 provides adaptability to send an extra BRM cell with a probability $p$ beside the periodic switch congestion check. The higher the threshold is, the faster the transient response is.

Consolidation noise is present in algorithm 2. All algorithms 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 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. Algorithm 4 may suffer from little noise because of its sensitivity to any bottleneck change.

As for RM cell ratio, algorithm 1 ensures that the ratio is one as time tends to infinity. Algorithms 2 and 3 have no limit for the ratio. Algorithm 4’s ratio converges quickly to one since it sends BRM cell at every N BRM cells received at most. Algorithm 5 too ensures the ratio is one over the long run.

4. Concluding Remarks

In the high speed ABR multicasting scenarios in ATM networks, how to control feedback information from different leaves and effectively use available bandwidth becomes a critical issue for the network performance. In the point-to-multipoint situations, feedback implosion is always among the most important problems to be solved. To deal with this problem, an efficient method is to consolidate feedback from downstream leaves at branch points.

In this paper, the performance of various feedback control schemes has been studied by simulation on two different network configurations. Feedback schemes are incorporated in branch points (switches) with the rate allocation algorithms. The criteria used for performance are the complexity, overhead, consolidation noise and RM cell ratio. In terms of complexity, algorithms 1, 2 and 3 are less complex as compared to algorithms 4 and 5. Algorithm 1 has a fast response for high threshold value, algorithm 3 exhibits slow transient response if threshold is close to zero, algorithm 4 is threshold independent and algorithm 5 has faster transient response for higher thresholds. Algorithm 2 has more noise than algorithm 4. Algorithm 1 ensures that the RM ratio is one as time tends to infinity, algorithms 2 and 3 have no limits for the ratio, algorithm 4’s ratio converges quickly to one and algorithm 5 ensures the ratio is one over the long run.

References