CHAPTER 4
THE PROPOSED ALGORITHM

A number of algorithms have been proposed for extending ABR congestion avoidance algorithms 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. 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.

In this chapter, we propose an improved algorithm for feedback consolidation, which combines benefits from the previous algorithms. We also propose a new RM ratio control method. The performance of the proposed algorithm and previously presented algorithms are compared under a variety of conditions.

4.1 INTRODUCTION

Among all previous proposed consolidation schemes, “the wait-for-all” scheme [15] 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. [18] had proposed an algorithm (discussed in section 3.3.6.3 of the previous chapter) that returns back a BRM cell not only when news are heard from all branches but also whenever a severe overload occurs. 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. 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 (discussed in section 3.3.8 of the previous chapter) to send an extra BRM cell which provides more flexibility to span the speed-overhead spectrum. But the algorithm lacked two important issues. First, it didn’t account for the local switch congestion state. Second, 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.

4.2 THE PROPOSED ALGORITHM

Here, we propose a new consolidation algorithm that combines the benefits of the above algorithms. In the new algorithm, there are two ways for the switch to trigger sending a BRM cell:

1- It returns a BRM cell if feedback is received from all branches, thus preserving the advantage of the “wait-for-all” scheme (accurate feedback).

2- An extra BRM cell is passed to the source if 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 4.1.

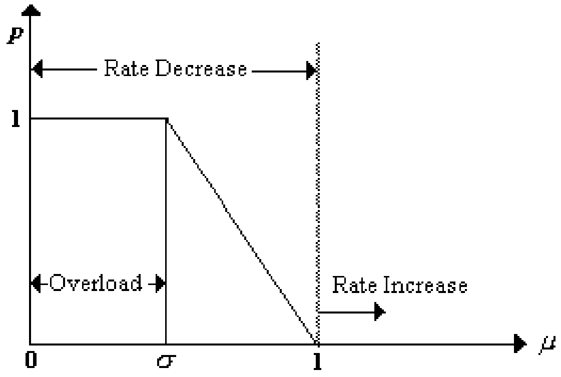


Figure 4.1: The probability function used by the proposed algorithm to send an extra BRM cell

μ denotes the ratio of the current MER and the rate indicated in the last returned BRM cell. 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 ($\sigma \le \mu \le 1$) 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 4.1.

Since the algorithm sends the extra BRM cells with a probability *p* if $\sigma \le \mu < 1$ (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 [21]. 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 algorithm, the rate allocation algorithm is performed whenever a BRM is received and not just when a BRM is being sent. Doing this, 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. First, in the steady state, the rate allocations tend to be stabilized. Second, the new FRM cell may carry a new 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.

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.

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.

We can conclude that the proposed algorithm has the following main features:

- It gives accurate feedback due to feedback synchronization.
- It takes the congestion state of local switch into consideration *with reduced overhead*.
- It detects the *severe overload* situations and sends immediate rate decrease feedback to the source leading to a fast overload transient response.
- In case of *moderate overload* situations, there is a chance to send an extra BRM cell, hence alleviating the threshold-sensitivity problem.
- 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 an FRM cell has been received by the branch point after 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 in Chen et al. algorithm. 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 [23], EPRCA [24], and Phantom [25]. We do not assume that a specific rate allocation algorithm is used in the switch. The algorithm operates as follows.

```
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 SendBRM = 0, Reset = 1
2. IF (NOT BRMReceived$_i$) THEN              // Set the flag of branch i
      A. Let BRMReceived$_i$ = 1
      B. Let 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
      A. Let SkipIncrease = SkipIncrease – 1  // An underload situation
      B. Let NBRMsRecv = 0
      C. 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, Let 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, Let 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
            2. Let NBRMsRecv = 0
            3. 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
```

4.3 THE PROPOSED RM RATIO CONTROL METHOD

Most of the previous algorithms control the RM ratio using a counter register, which 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 counter register is greater than zero, this particular feedback can be ignored and counter decremented. Thus, a feedback may be ignored even if the number of sent BRM cells is less than the number of received FRM cells.

Here, we propose a new method to control the RM ratio. The new method depends on the difference between the FRM cells received and the BRM cells sent by the switch. A positive difference indicates that the switch can send more BRM cells. At the transient period, this difference increases (as the source sends FRM cells but no feedback is received yet). The switch benefits from this difference to send more BRM cells. If the difference is negative then an underload feedback should be ignored. The new method exhibits better RM ratio (converges faster towards one) if there is a highly change of bottleneck values in the network or when the probability to send extra BRM cells is high.

This new method is very simple and straightforward to implement. It could be used by any consolidation algorithm that permit sending extra BRM cells in overload cases.

A register, FRMminusBRM is the new controller of the RM ratio as mentioned above. FRMminusBRM is incremented whenever an FRM cell is received and decremented whenever a BRM cell (extra or not) is sent.

The proposed algorithm using the new RM ratio control method operates as follows.

```
Upon the receipt of an FRM cell:
1. Multicast FRM cell to all participating branches
2. Let AtLeastOneFRM = 1
3. Let FRMminusBRM = FRMminusBRM + 1      // New RM ratio controller

Upon the receipt of a BRM cell from branch i:
1. Let SendBRM = 0, Let Reset = 1
2. IF (NOT $BRMReceived_i$) THEN             // Set the flag of branch i
      A. Let $BRMReceived_i$ = 1
      B. Let 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(FRMminusBRM $\leq$ 0)AND(NBRMsRecv = Nbranches)) THEN
      A. Let NBRMsRecv = 0                // An underload situation
      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
   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
6. IF (SendBRM) THEN
      1st.  Pass the BRM with ER = MER to the source
      B. Let FRMminusBRM = FRMminusBRM - 1   // New RM ratio controller
      C. IF (Reset) THEN
            1. Let MER = PCR
            2. Let NBRMsRecv = 0
            3. 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
```

4.4 PERFORMANCE RESULTS & ANALYSIS

This section provides a selected set of results, obtained using simulation. These results compare the performance of our proposed algorithm to algorithms from the literature.

We first discuss the simulation model used, then compare the algorithms in a variety of configurations with and without constant bit rate (CBR), variable bit rate (VBR) background, and with various link lengths, bottleneck locations, and number of leaves.

In the results that follow, “Fahmy” denotes Fahmy et al. algorithm, “Chen” denotes Chen et al. algorithm, “Ros” denotes Ros et al. algorithm, ”Modified-Ros” denotes Ros et al. algorithm using the new RM ratio control scheme, “Proposed” denotes our proposed algorithm using Fahmy’s RM ratio control scheme, and “Modified-Proposed” denotes our proposed algorithm using the new proposed RM ratio control scheme. Two graphs are plotted for each configuration: the allowed cell rate and the BRM/FRM ratio for the point-to-multipoint ABR source. A third graph for the queue length is added only for overloaded switches. We combine the graphs; of a certain measure of performance for the six algorithms, in one figure for the simplicity of comparison. We simulate the six algorithms with two different thresholds: low (0.05), and high (0.95) for each configuration.

4.4.1 The Simulation Model

4.4.1.1 Simulation Approach

We have used the “event-scheduling” simulation approach. This approach is based on the concept that actions may be taken only when one of the events takes place. At any other point in time, no change occurs in the system’s measures of performance, and hence the system should be left alone.

The general idea of this approach is that whenever a new event is generated, it is automatically placed in its proper chronological order on the time scale. The simulation process will then select the first event that takes place and depending on its type will perform the appropriate actions. Such an event will then be removed from the list indicating that the event has been fathomed. This process is repeated until the desired simulation period is covered.

4.4.1.2 Switch Model

A switch interconnects multiple links and supports multiple ports, typically an input port or/and an output port per-link. Each port may have some buffers associated with it. It is possible to put the buffers exclusively at the input port (an input-buffered architecture), exclusively at the output port (an output-buffered architecture), or at both the input and output ports. Popular switch architectures tend towards being exclusively output buffered due to its superior performance when compared to input buffered switches [11]. We choose to focus on output buffered switch architectures. Buffers may be logically partitioned into queues, which are scheduled using a specific discipline. Queuing and scheduling at the buffers may be handled in a First In First Out (FIFO) manner where all the cells coming to the port are put into a common buffer (and later serviced) in the order they arrived at the port. A switch will have a separate FIFO queue for every traffic class supported (CBR, VBR, ABR, and UBR classes). The rate-based framework defined in the ATM Traffic Management 4.1 [4] standards allows the switch designers total flexibility in choosing the buffer allocation, queuing, and scheduling policy. This was one of the key features that led to its acceptance compared to the credit-based proposal which required per-VC queuing and scheduling to be implemented at every switch. We assume a model of an output buffered switch implementing per-class queues at every output port. The ABR congestion control algorithm runs at every output port's ABR queue.

The capacity of the output link is assumed to be shared between the “higher priority” classes (constant bit rate (CBR), real-time variable bit rate (rt-VBR), and non-real time variable bit rate (nrt-VBR)) and the available bit rate (ABR) class. Link Bandwidth is first allocated to the CBR class then to the VBR class and the remaining bandwidth, if any, is given to the ABR class traffic.

4.4.1.3 Validation

Validation is concerned with determining whether the conceptual simulation model is an accurate representation of the system under study. Our simulation model is validated using two techniques:

1. Structured Walk-Through: As a core step of our model validation, a structured walk-through is what we really did. The most important points of our simulation conceptual model are traced to ensure that the model's assumptions are correct, complete and consistent.
2. Comparing Output Results: Our output results are compared with the existing data and the two sets relatively coincided.

4.4.1.4 Verification

Verifying a simulation program is to make sure that it represents the simulation model it is intended to represent. We used three techniques to verify our simulation program:

1. In the programming phase, we worked in steps to write and debug the simulation program. The steps are:

   a- Writing the main data structure and testing it.

   b- Writing events and debugging the program.

   c- Writing the statistics code and debugging the program.

2. Running the simulation under a variety of settings of the input parameters and checking the output to see that it is reasonable. As the output was compared with the results in [18,26] and was verified, using simple measures of performance.

3. Tracing the state of the simulated system, i.e., the events list, variables, statistical counters, etc., and printing them is a log file, to see if the program is operating as intended. It was taken into consideration to trace all the program paths and to test the extreme conditions by entering special data for input.

4.4.2 Parameter Settings

Throughout our experiments, the following parameter values are used:

- All links have a bandwidth of 155.52 Mbps.
- All P-to-MP 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 P-to-P connections.
- All sources are deterministic, i.e., their start/stop times and their transmission rates are known. 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 Ros et al. algorithm is set to the minimum of 2 and number of branches of the multicast connection at the switch.

4.4.3 Simulation Results

4.4.3.1 Parking Lot Configuration

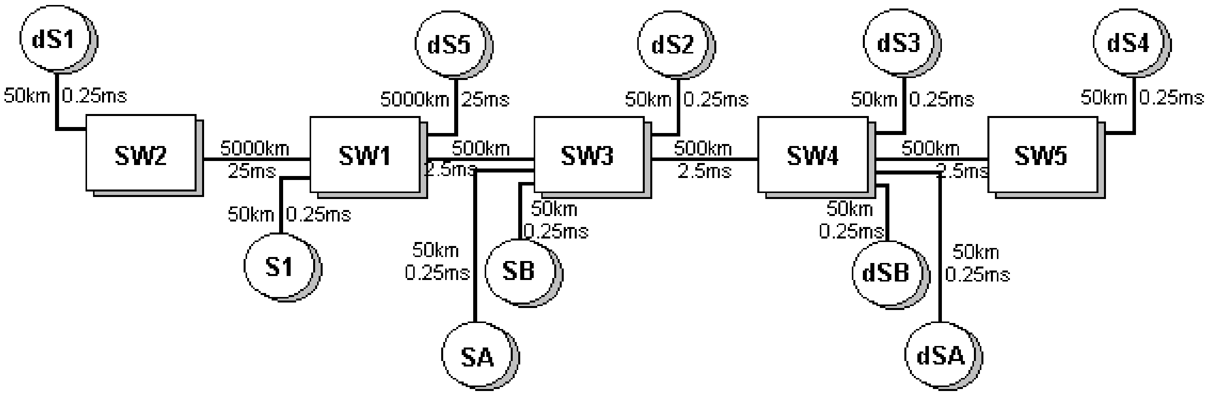


Figure 4.2: Parking Lot Configuration

The configuration is shown in figure 4.2 [21]. The configuration 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 = (0.9*155.52)/2 $\approx$70 Mbps before 100 ms. The rate decrease ratio, μ is $\approx$0.5 for the multicast connection.

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 = (0.9*(155.52-90))/2 $\approx$29.5 Mbps. The rate decrease ratio, μ is $\approx$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.

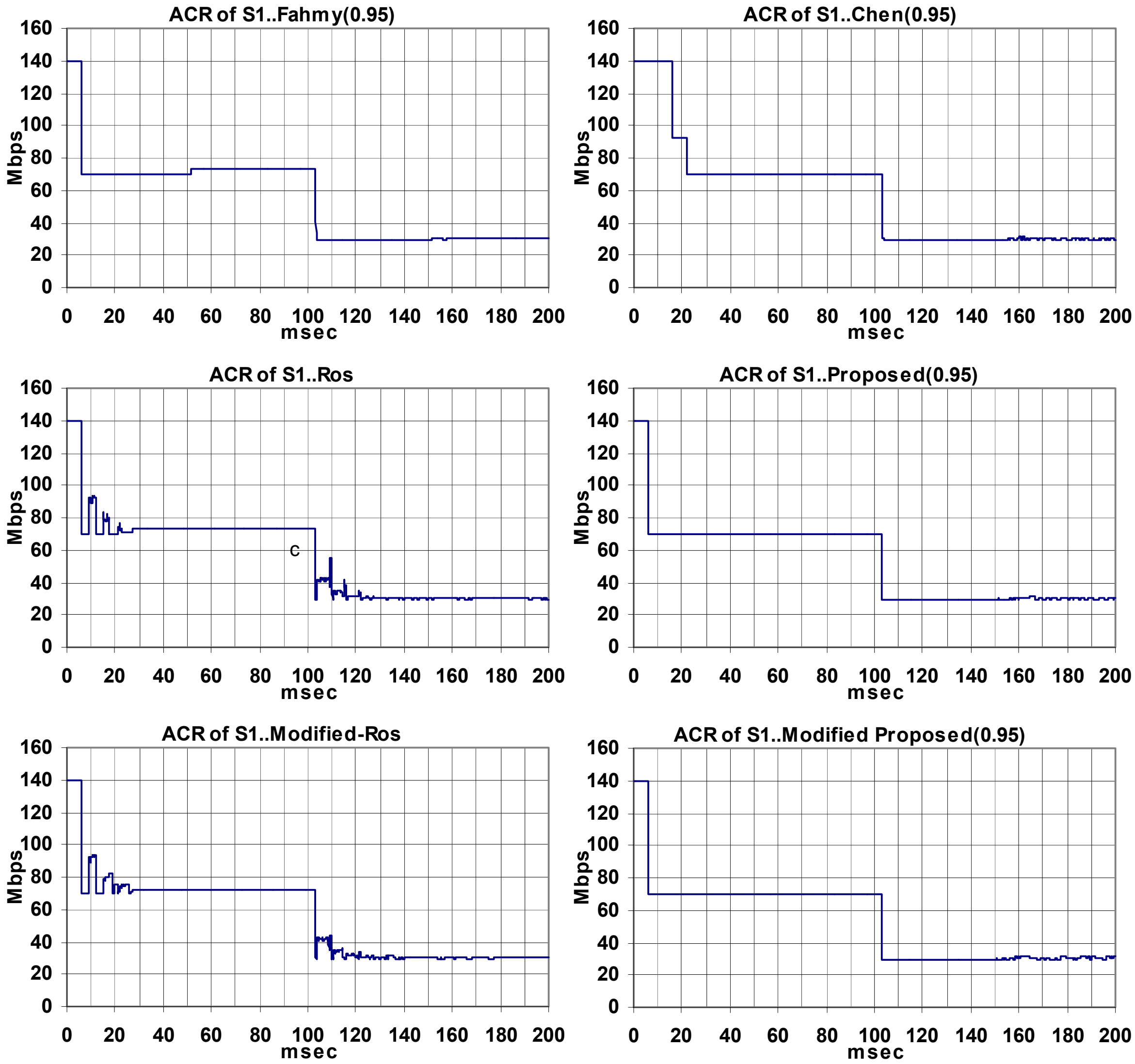
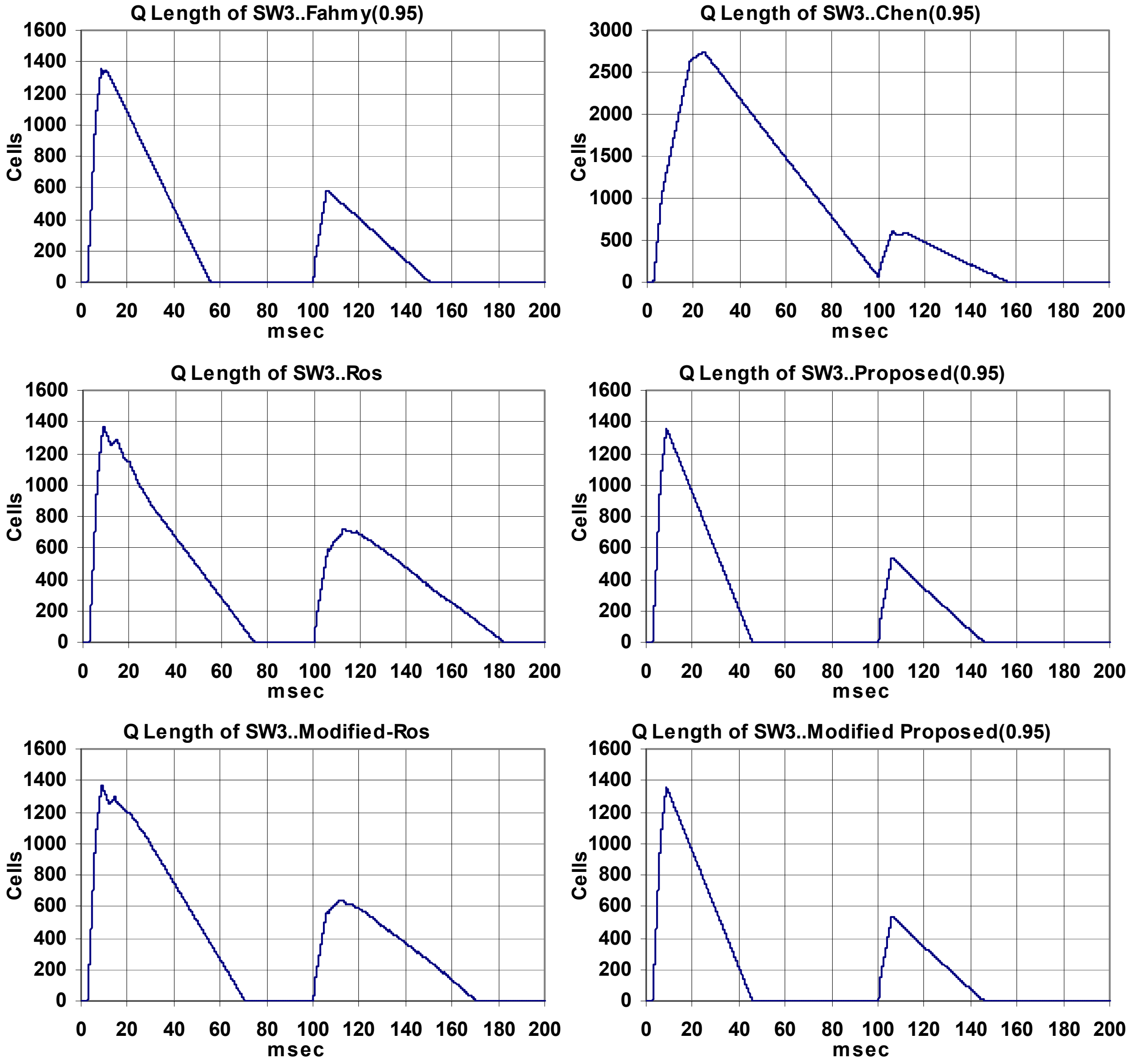


Figure 4.3: Allowed Cell Rate in Mbps for P.L. Config. with high threshold

The ACR graphs (shown in figure 4.3) for Fahmy, Proposed and Modified-Proposed algorithms indicate the same fast transient response during both phases due to the periodic switch congestion check. All 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 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 the change of the bottleneck value. Note that the other algorithms store only the minimum rate until the feedback is received from all the branches at the time the point-to-point algorithm has been converged to the optimal value. Ros-Modified algorithm exhibits the same performance as expected.


Q Length of SW3..Fahmy(0.95)

Q Length of SW3..Chen(0.95)

Q Length of SW3..Ros

Q Length of SW3..Proposed(0.95)

Q Length of SW3..Modified-Ros

Q Length of SW3..Modified Proposed(0.95)

Figure 4.**4:** Queue length of the bottleneck link for the P.L. Config. with high threshold

All the six algorithms detect the second rate drop in phase 2 quickly because the feedback news becomes available in the network.

Due to the late response of Chen algorithm, initial queue of the bottleneck link is larger than the other algorithms. This is shown in figure 4.4. The queue draining in case of Ros algorithm is slower than the others because of the noise rates.

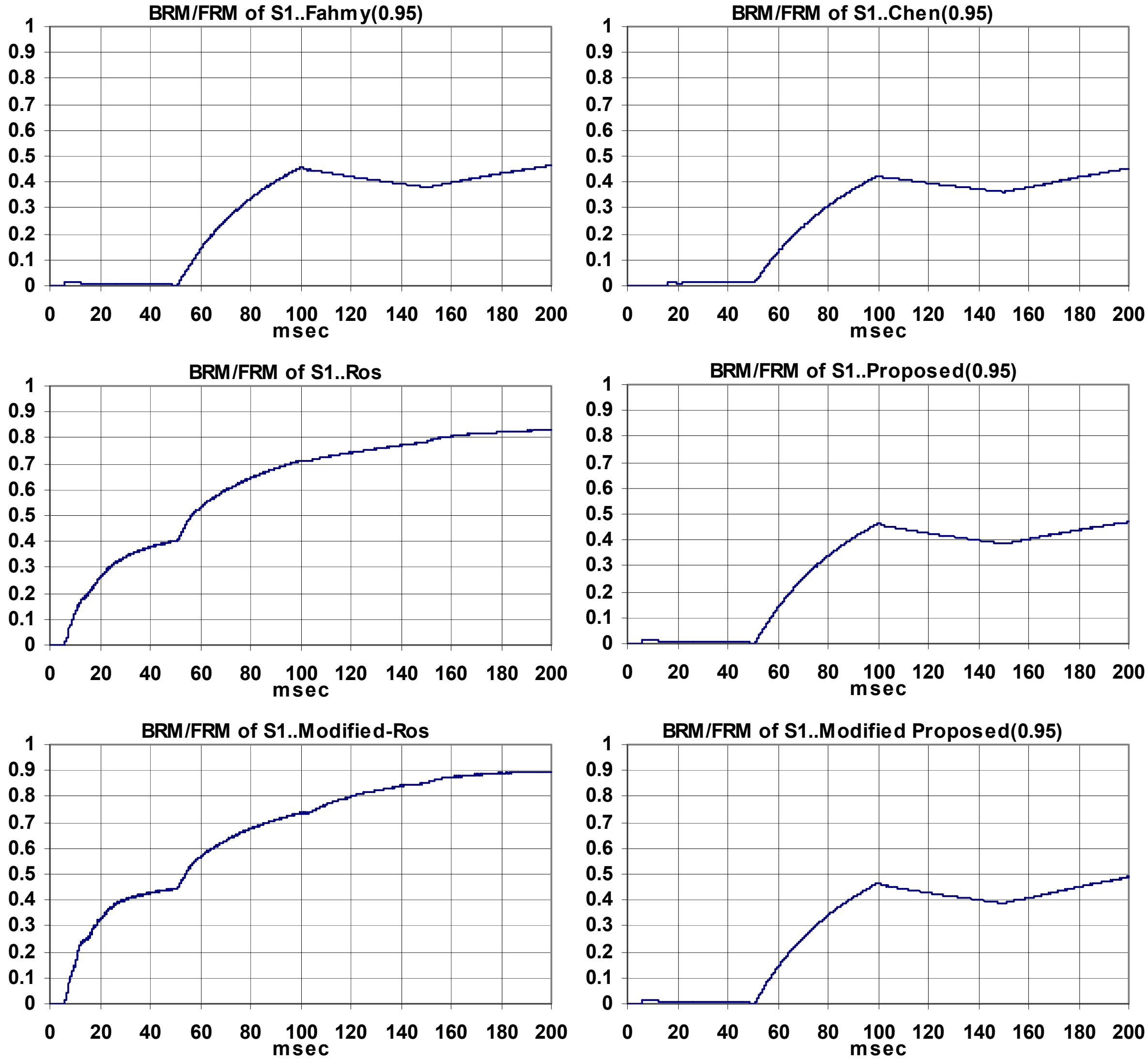


Figure 4.5: The RM ratio of the source for the P.L. Config. with high threshold

Figure 4.5 indicates no increase in the RM ratio of the Modified-Proposed algorithm to that of the Proposed one. While, a remarkable increase is shown in Ros-Modified algorithm than in Ros algorithm. This is due to the relatively large rate of change of the feedback values in Ros algorithm and the periodic sending of BRM cells.

Figures 4.6 through 4.8 illustrate the performance of the six algorithms with low threshold. Note that the graphs for Ros and Ros-Modified algorithms will not be changed since they are both threshold-independent.

Performance of Fahmy 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.

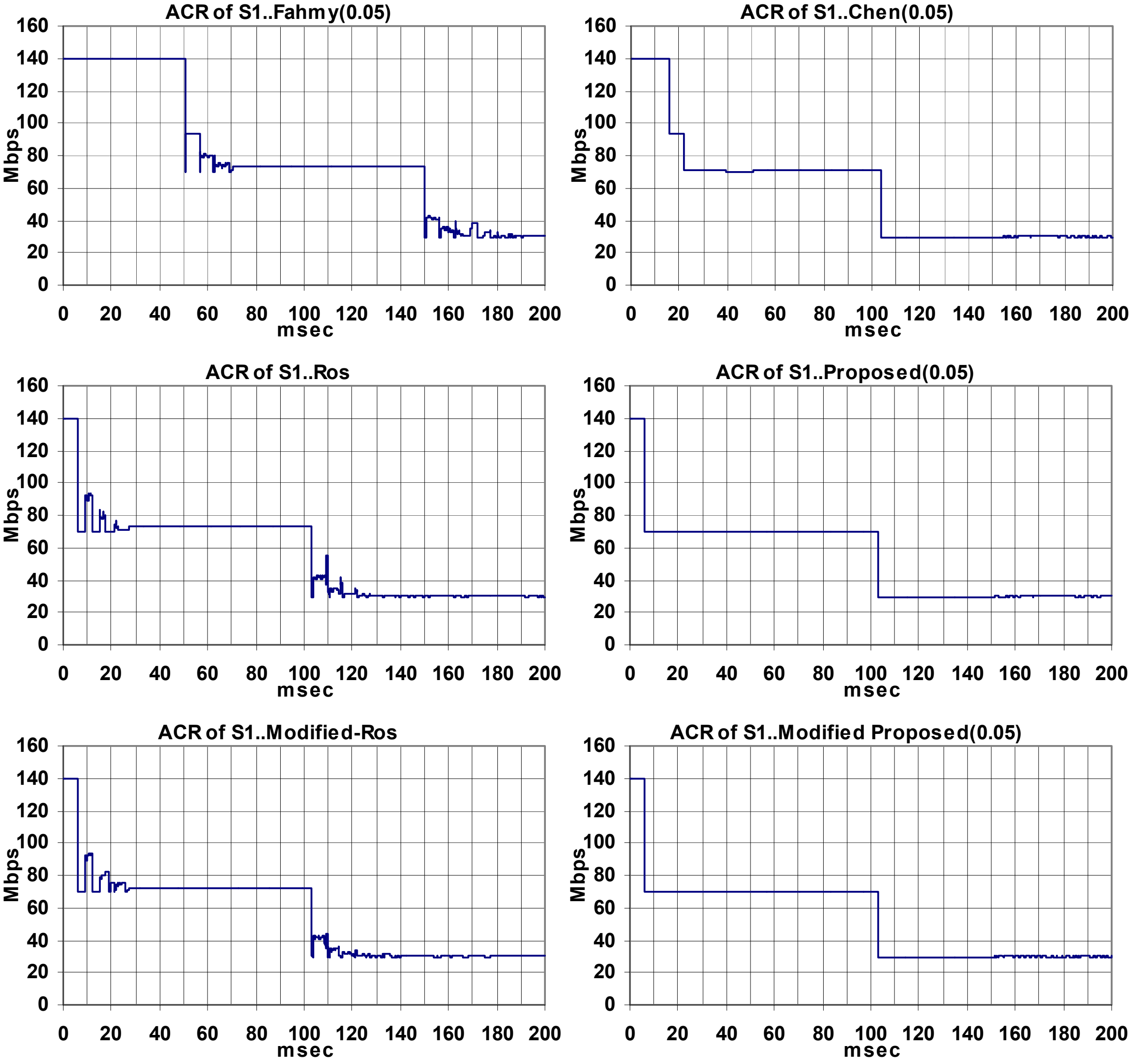
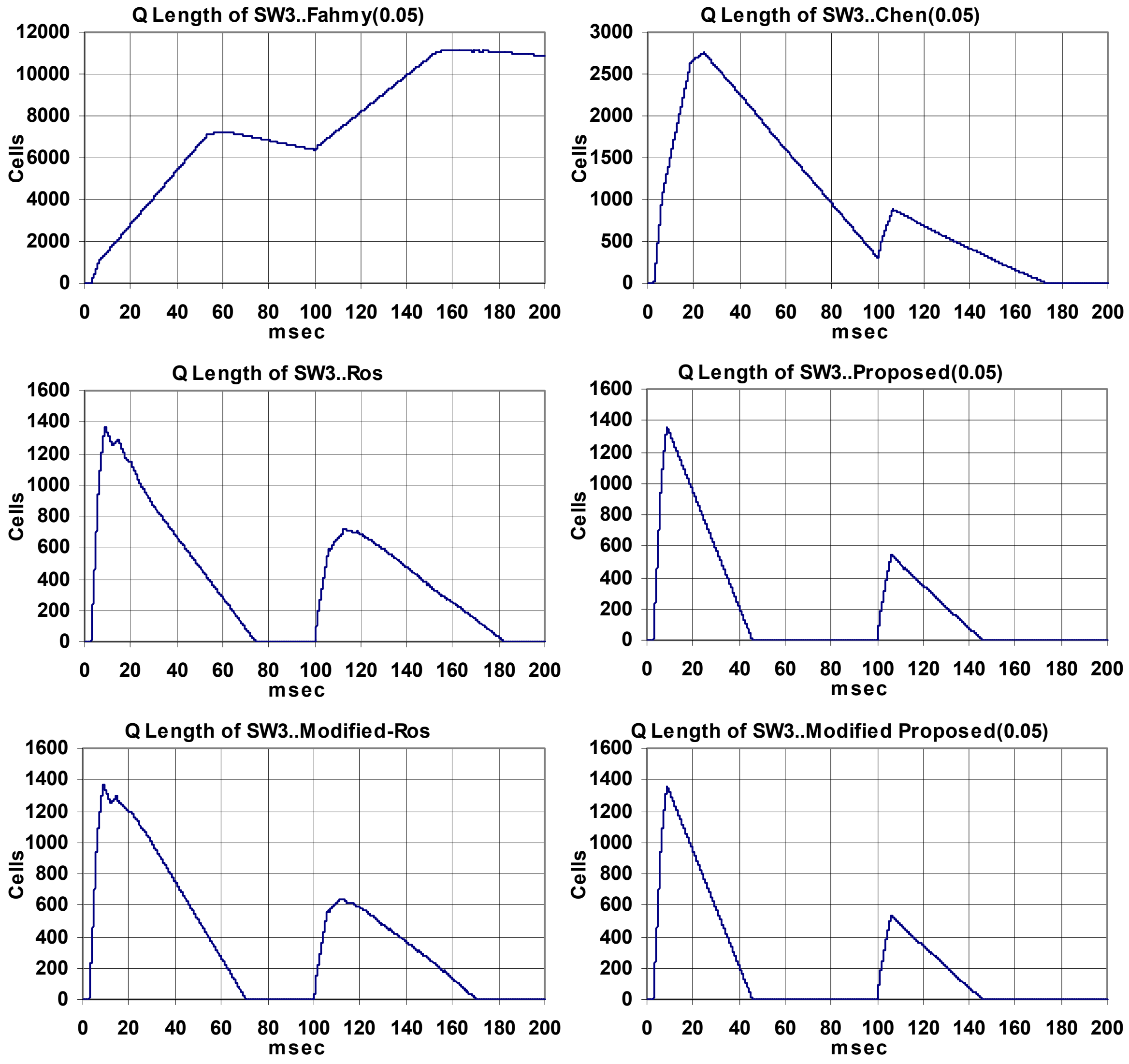


Figure 4.6: Allowed Cell Rate in Mbps for P.L. Config. with low threshold

Due to the same reason, the RM ratio in Fahmy algorithm is less than in Chen algorithm (because of the low chance to send BRM cell in Fahmy algorithm) contrary to the case with high threshold. Note that Chen algorithm doesn’t include any RM ratio control scheme.

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 (1-0.5)/(1-0.05) ≈0.53), which affect both transient response and queues length.



Q Length of SW3..Fahmy(0.05)

Q Length of SW3..Chen(0.05)

Q Length of SW3..Ros

Q Length of SW3..Proposed(0.05)

Q Length of SW3..Modified-Ros

Q Length of SW3..Modified Proposed(0.05)

Figure 4.7: Queue length of the bottleneck link for the P.L. Config. with low threshold

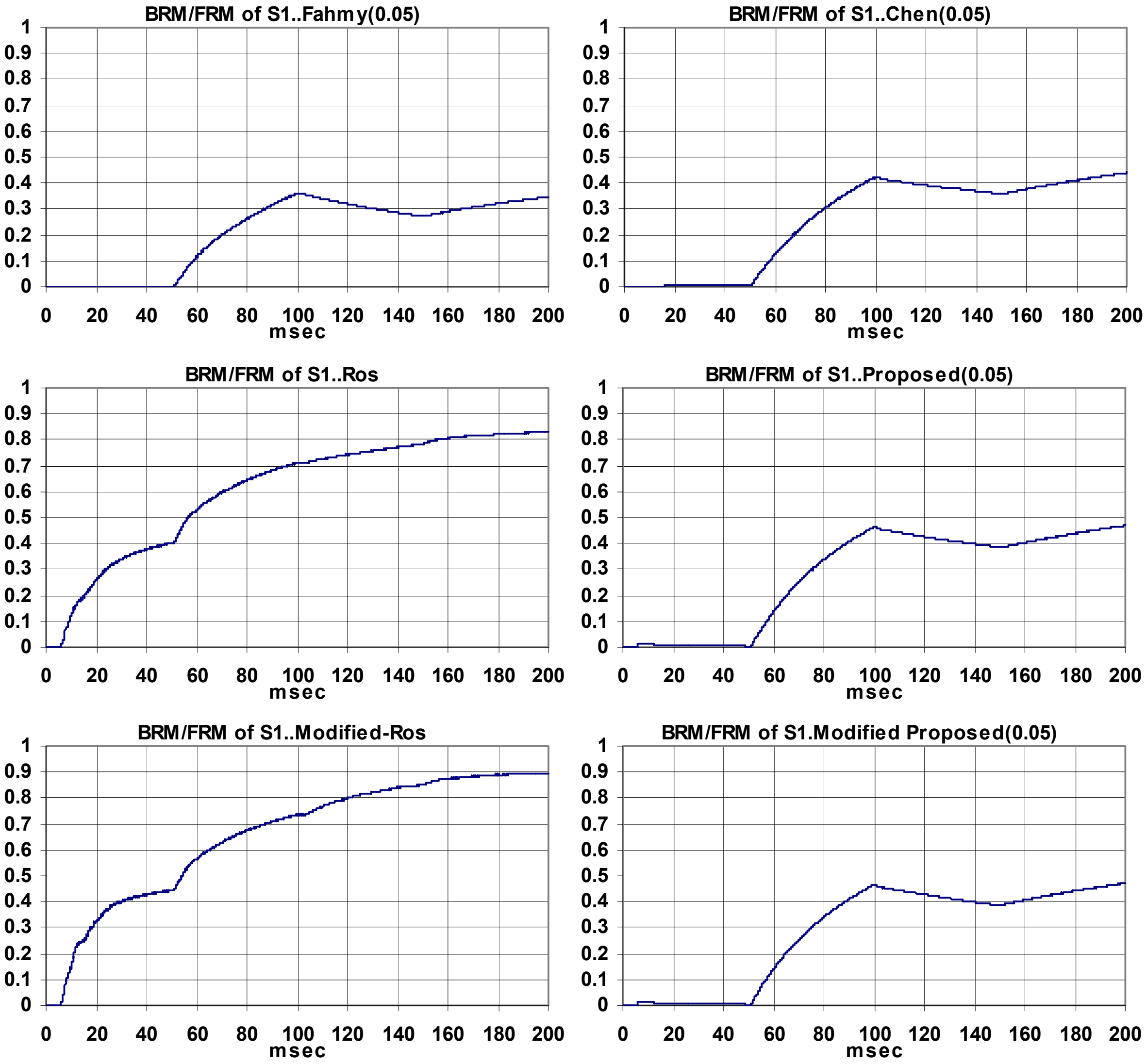


Figure 4.8: The RM ratio of the source for the P.L. Config. with low threshold

4.4.3.2 Chain Configuration

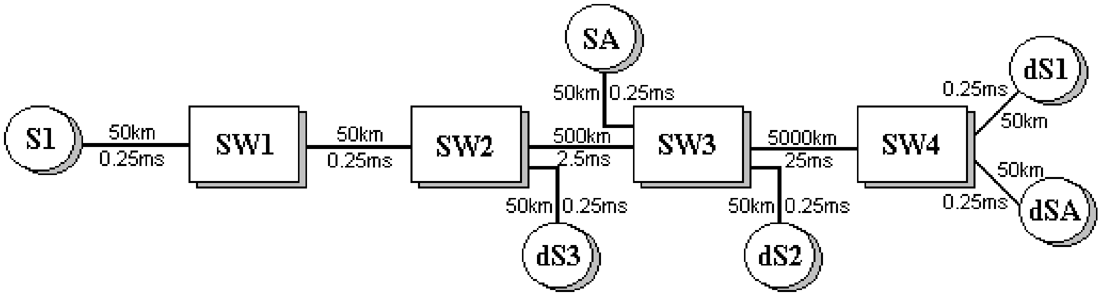


Figure 4.**9:** Chain Configuration

The chain configuration, illustrated in figure 4.9 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.

Fahmy et al. argued that his configuration is an ideal configuration for illustrating the consolidation noise problem [18]. Figures 4.10 through 4.12 illustrate the performance of the six algorithms with high threshold.

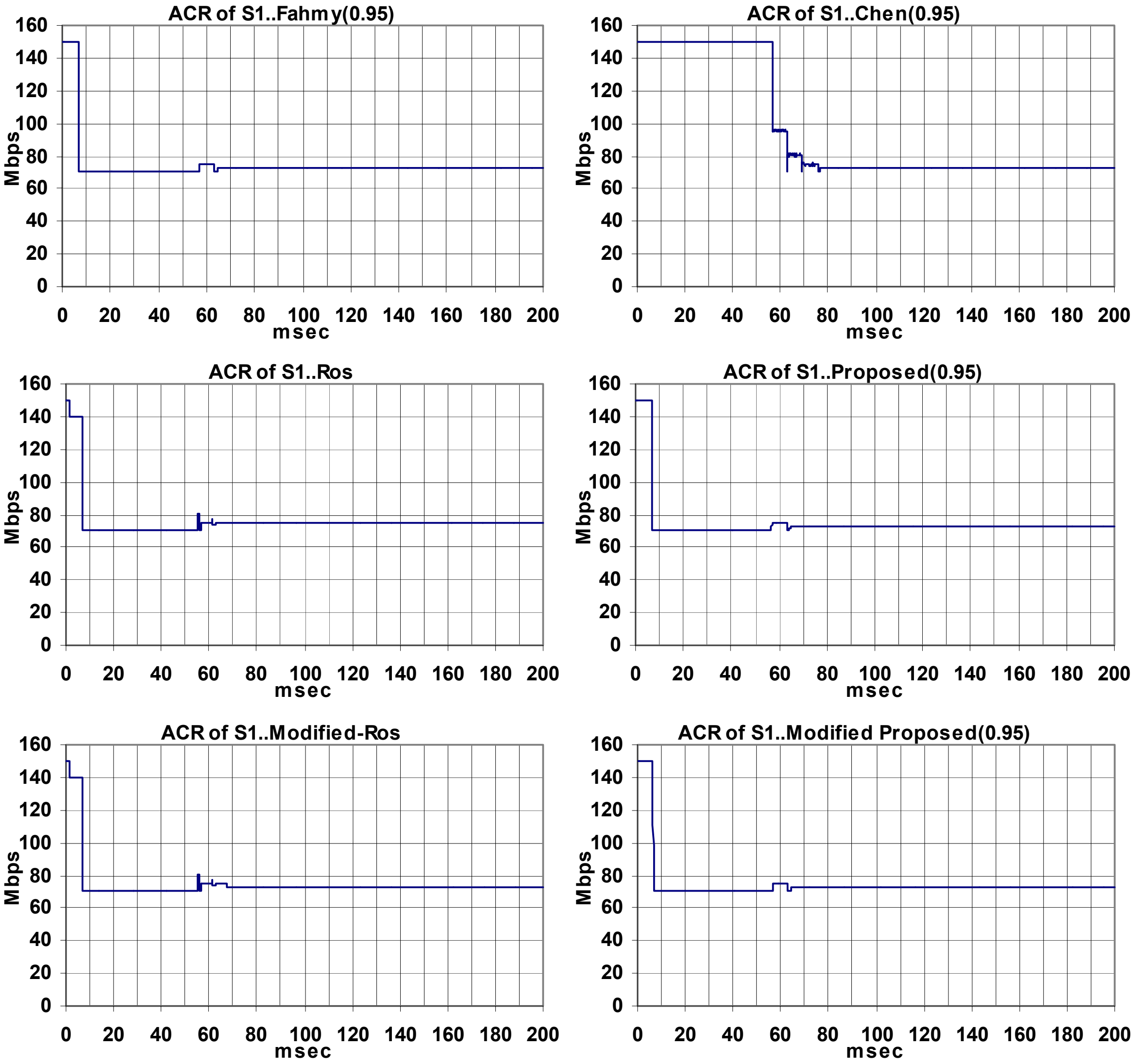


Figure 4.10: Allowed Cell Rate in Mbps for Chain Config. with high threshold

The bottleneck rate is (0.9*155.52)/2 = 70 Mbps and the rate decrease factor is 0.5. Fahmy, Proposed and Modified-Proposed algorithms yield optimal performance in this case 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 time from S1 to dS2). Chen algorithm suffers from slow transient response. The rate of S1 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 graphs of Ros and Ros-Modified algorithms is not detected by other algorithms since the rate decrease ratio is 1. This drop appears because ICR=150 (not 140) Mbps in this configuration.

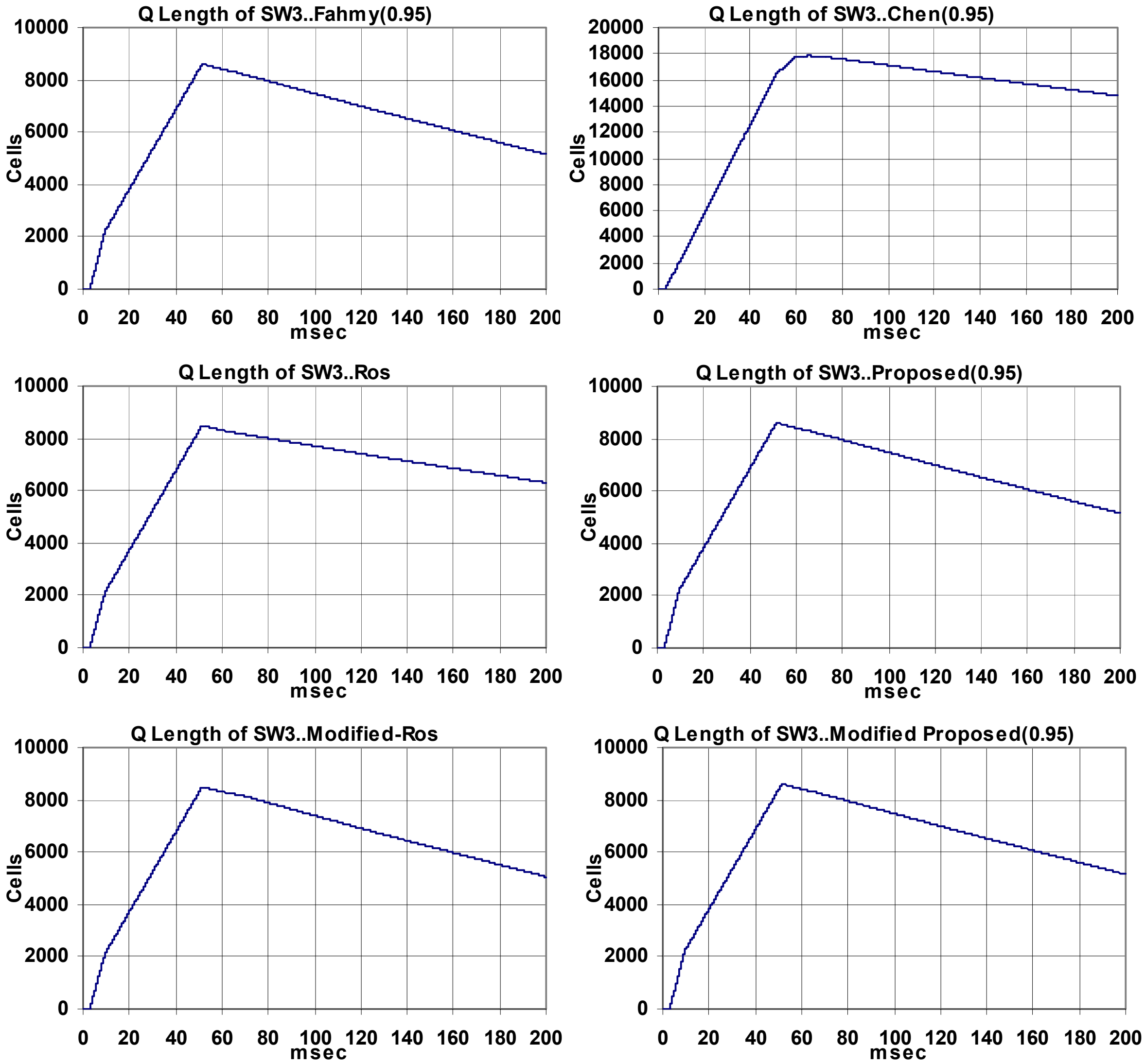


Figure 4.**11:** Queue length of the bottleneck link for the chain config. with high threshold

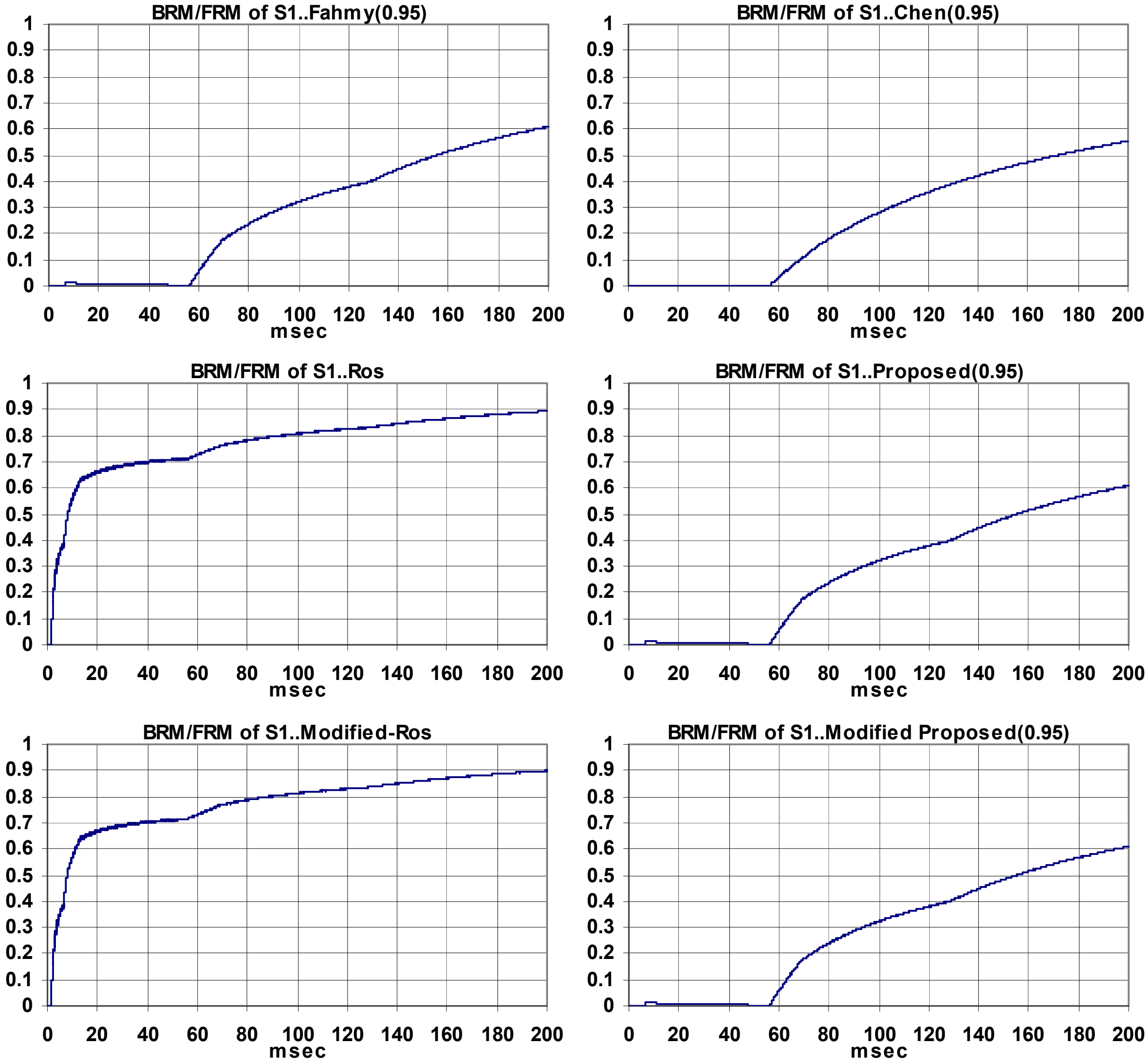
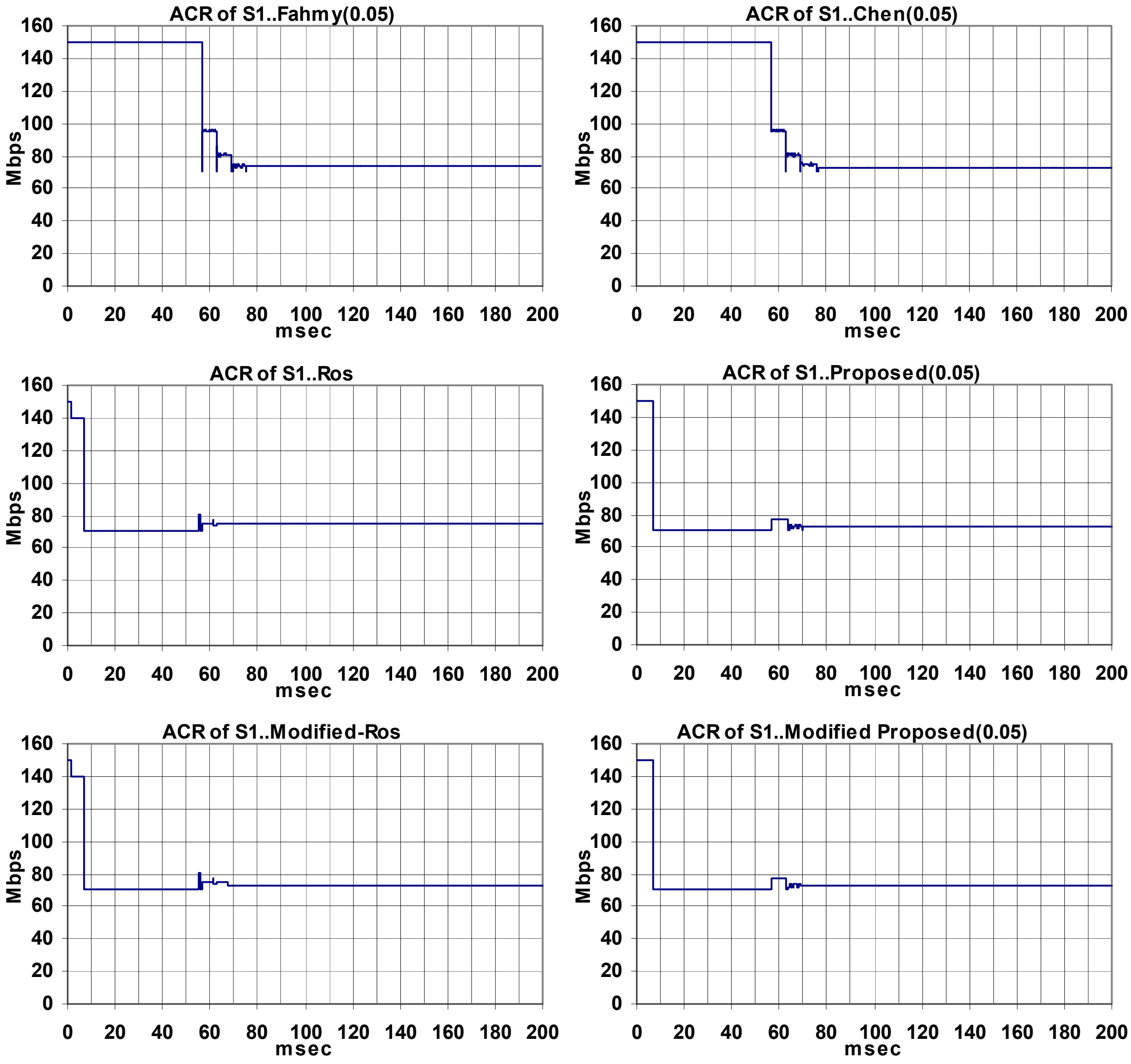


Figure 4.12: The RM ratio of the source for the Chain Config. with high threshold

A very slight improvement is indicated in RM ratio graphs when using the new RM ratio control scheme. This is expected as the network has reached the steady state quickly.

Figures 4.13 through 4.15 illustrate the performance of the six algorithms with low threshold. Here, 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 algorithms keep the same fast transient response as in case of high threshold. Here also, the effect of the new control scheme is very slight due to the large network stability period.


ACR of S1..Fahmy(0.05)

ACR of S1..Chen(0.05)

ACR of S1..Ros

ACR of S1..Proposed(0.05)

ACR of S1..Modified-Ros

ACR of S1..Modified Proposed(0.05)

Figure 4.13: Allowed Cell Rate in Mbps for Chain Configuration with low threshold

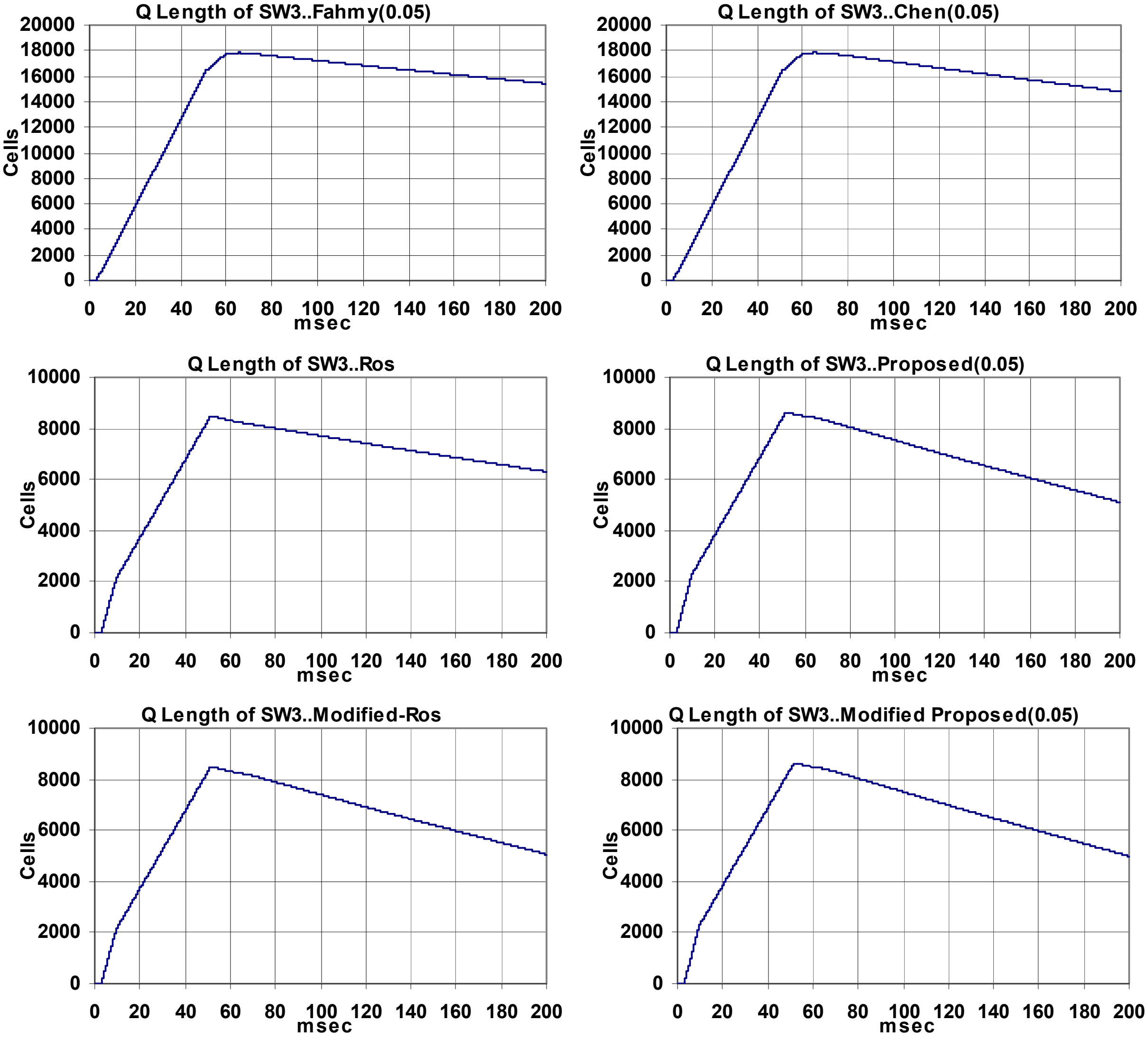


Figure 4.14: Queue length of the bottleneck link for the Chain Config. with low threshold

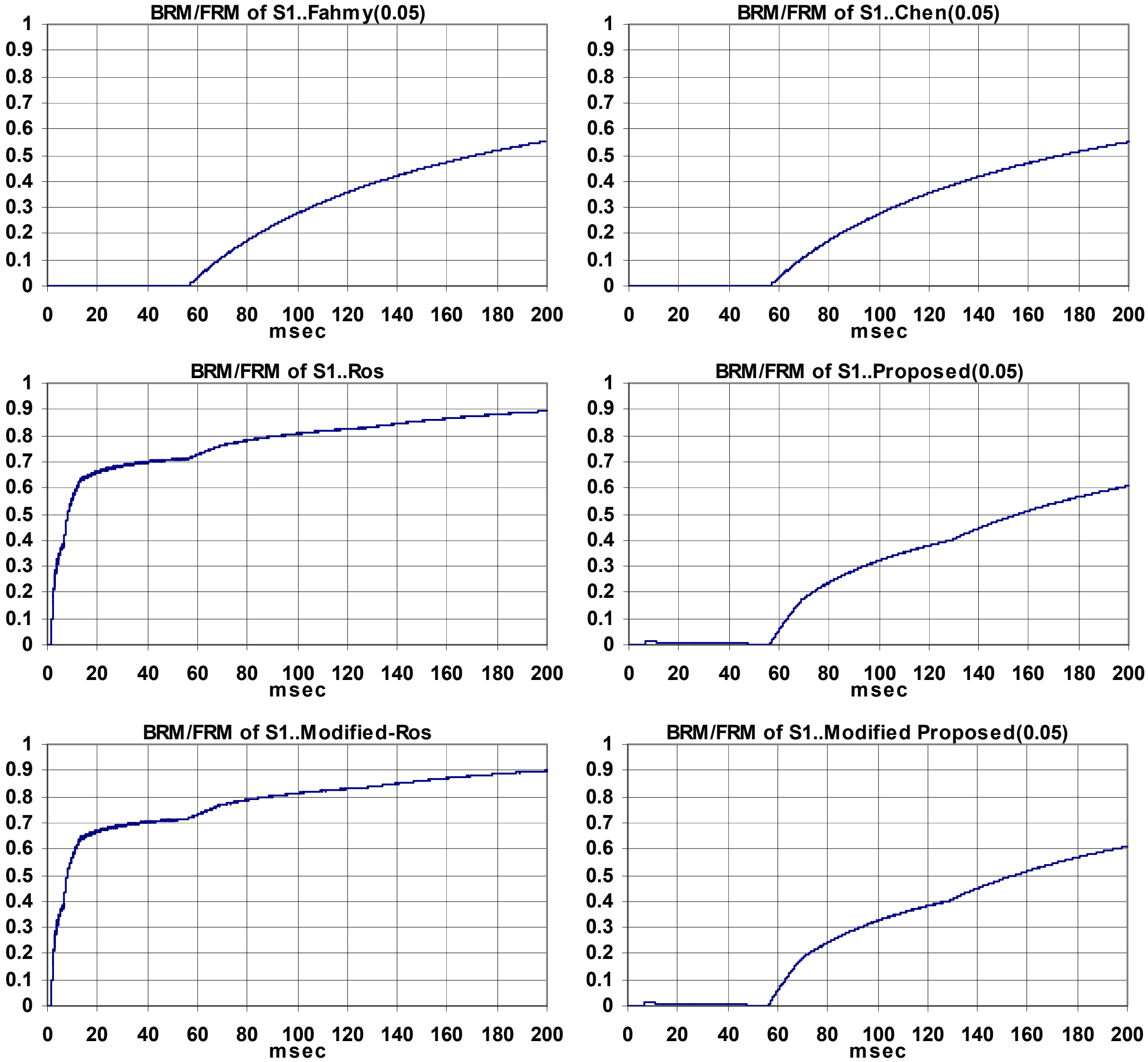


Figure 4.15: The RM ratio of the source for the Chain Config. with low threshold

4.4.3.3 Jumping-bottleneck Configuration

This configuration (shown in figure 4.16) is a modified version of the configuration in [22]. 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.

.

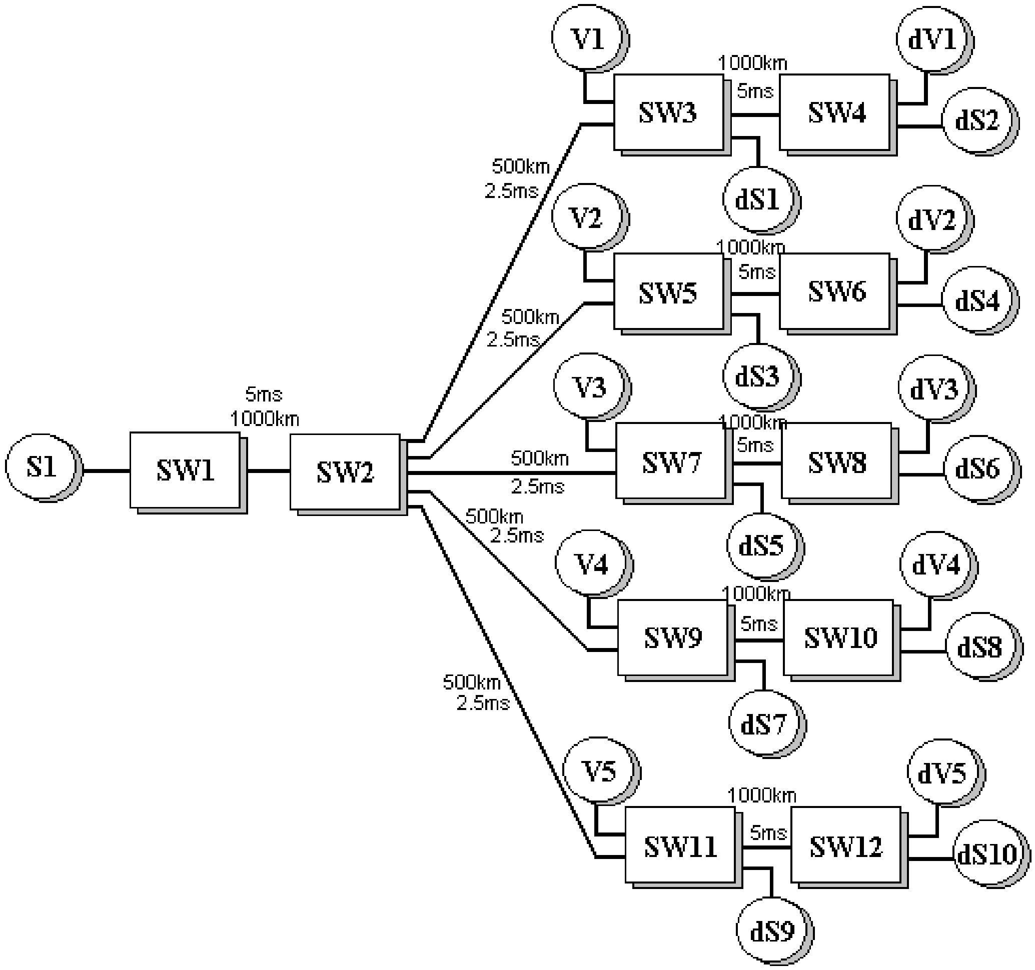


Figure 4.16: Jumping-Bottleneck Configuration

The sequence of events of the configuration in the transient period is shown in table 4.1. The table illustrates the event, the time of the event, the new bottleneck switch, and the new bottleneck rate after the event has been occurred. Note that the events from 2 to 11 will be repeated till the end of simulation.

Table 4.1: Events of transient period of Jumping-Bottleneck Configuration

Event No.	Time (ms)	Event	New Bottleneck Switch and Rate
1	0	V1 becomes active	SW3,135.5
2	8	V2 becomes active	SW5,108.5
3	16	V3 becomes active	SW7,81.5

4	20	V1 becomes inactive	SW7,81.5
5	24	V4 becomes active	SW9,59
6	28	V2 becomes inactive	SW9,59
7	32	V5 becomes active	SW9,59
8	36	V3 becomes inactive	SW9,59
9	40	V1 becomes active	SW9,59
10	44	V4 becomes inactive	SW11,104
11	48	V2 becomes active	SW11,104

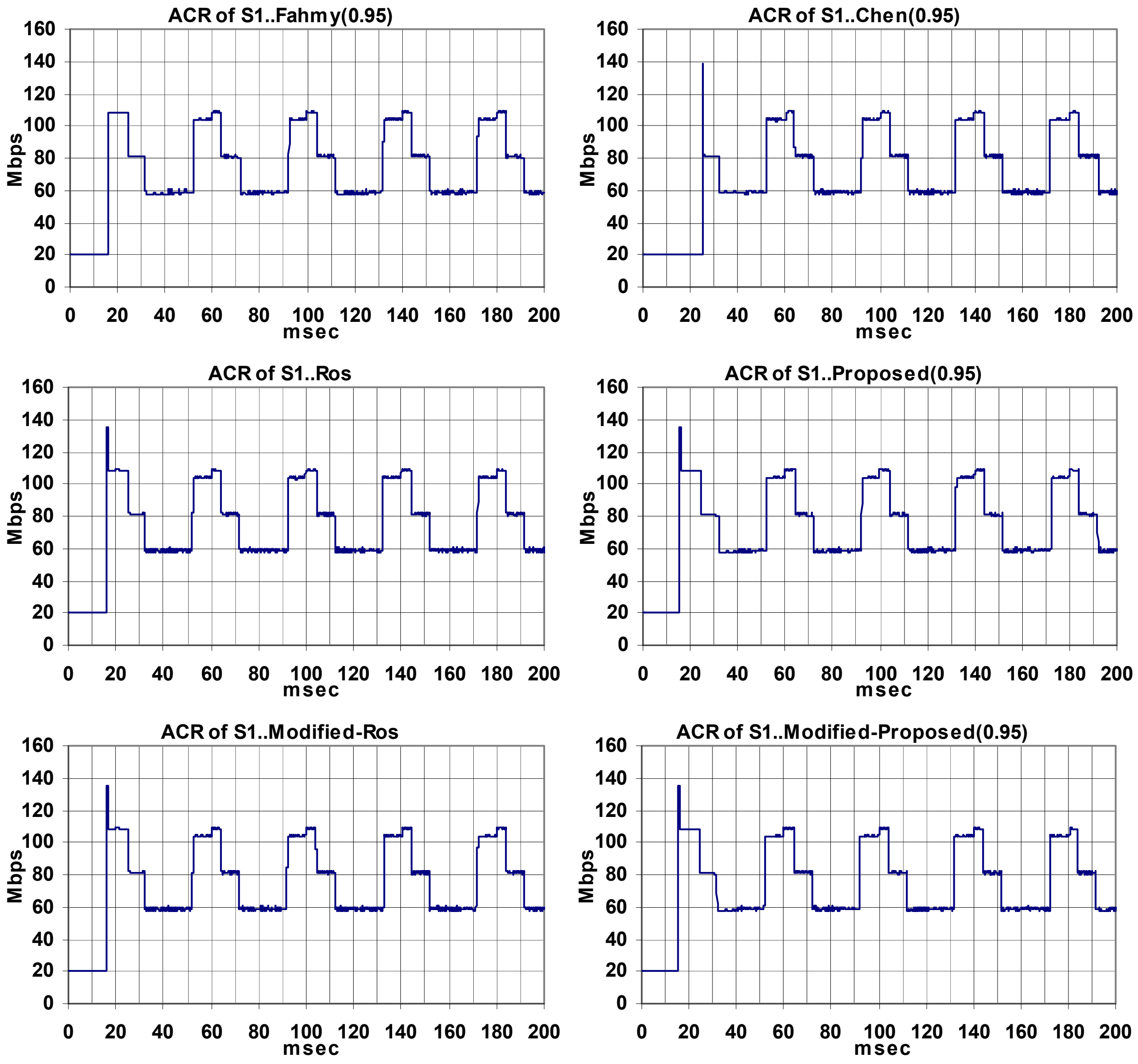


Figure 4.17: Allowed Cell Rate in Mbps for J.B. Config. with high threshold

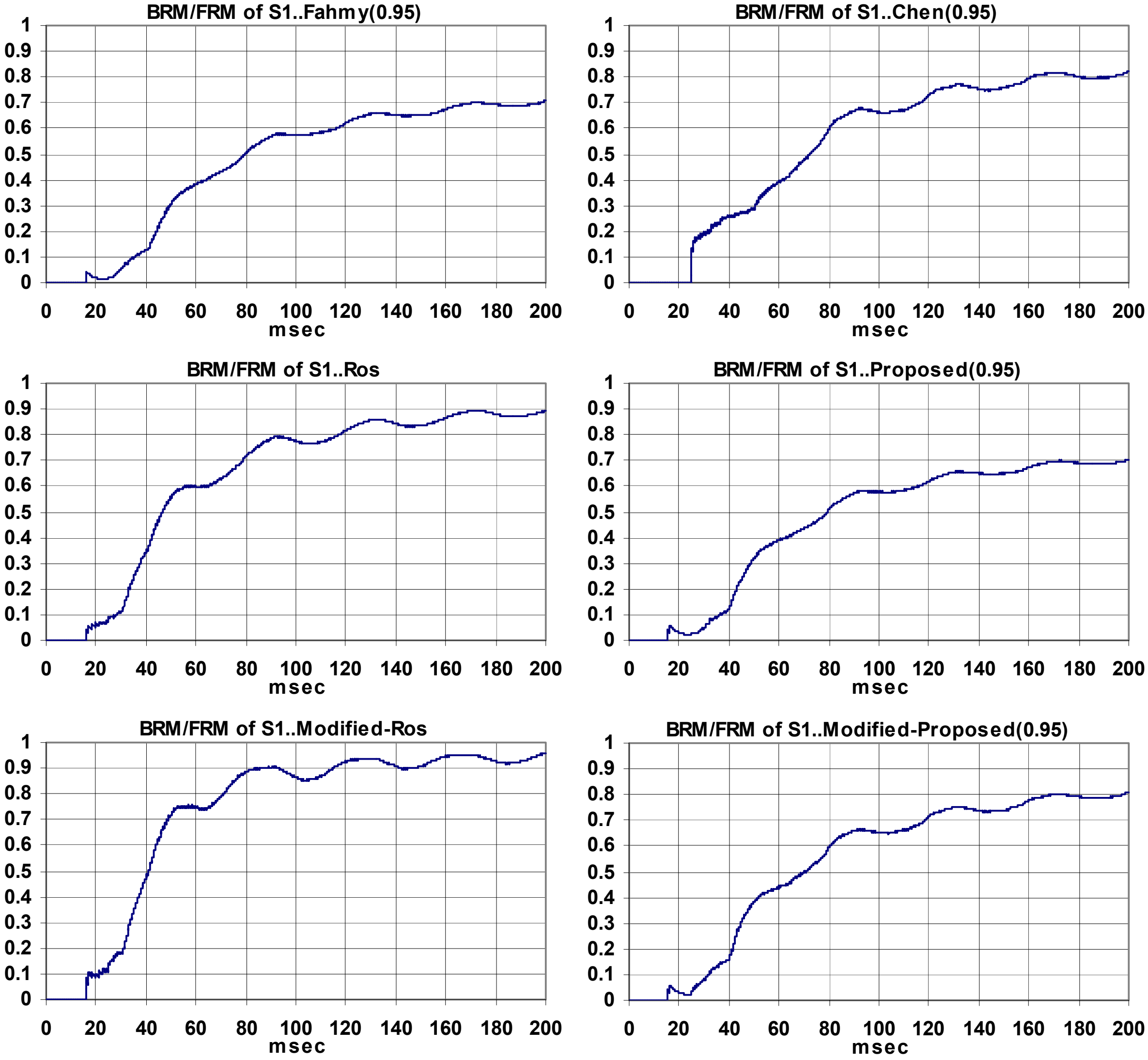
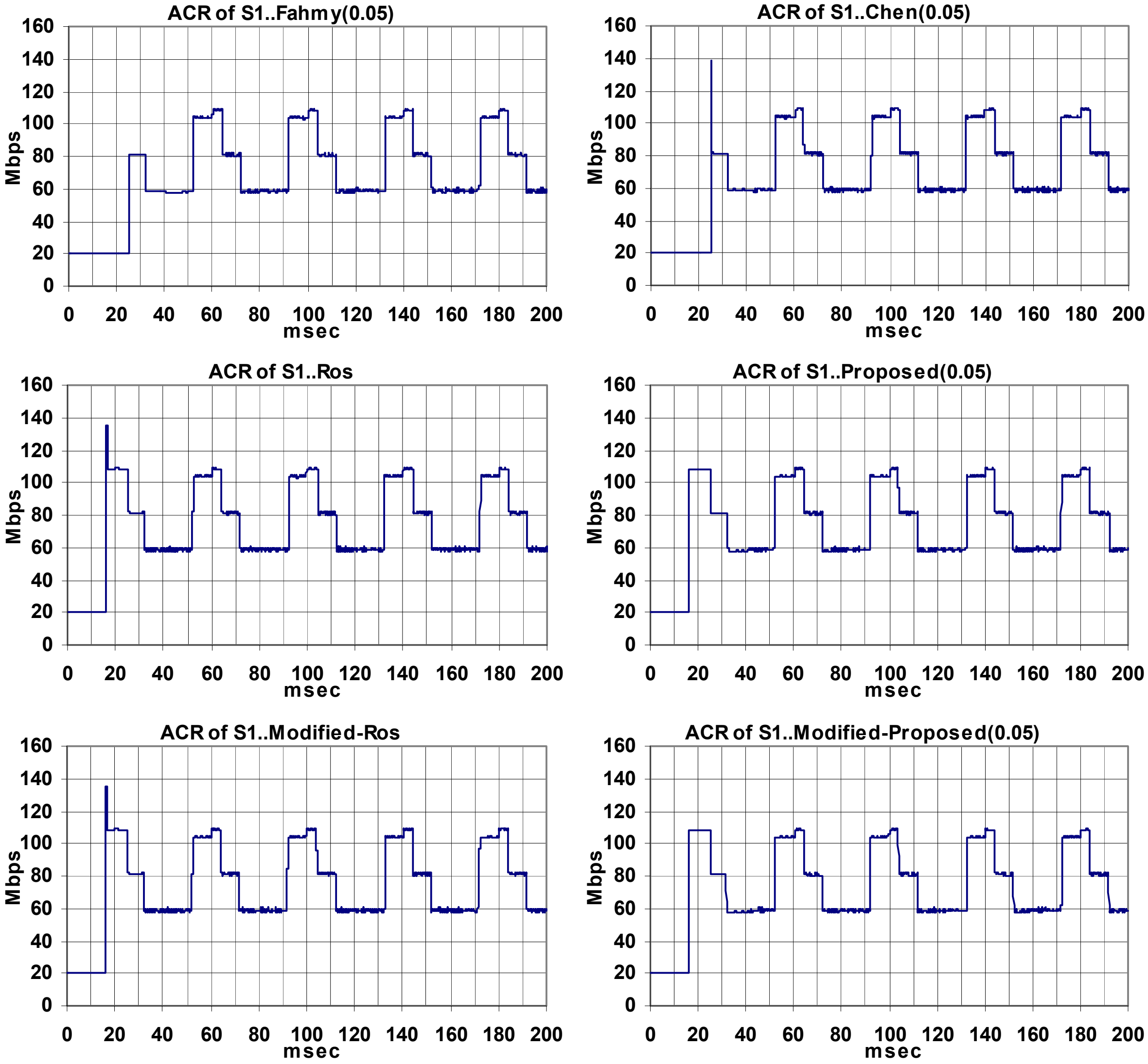


Figure 4.18: The RM ratio of the source for J.B. Config. with high threshold

Figures 4.17 and 4.18 illustrate the performance of the six algorithms with high threshold. The salient features of these graphs are as follows.

- 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 (not round-trip time) from the bottleneck switch to the source.
- Ros algorithm has an optimal performance since SW2 have the most 2 bottleneck values in hand all the time.

- Fahmy algorithm doesn’t detect the first peak since the rate decrease ratio is 135.5/140 ≈ 0.968, which is greater than 0.95, while the second peak is detected because the rate decrease ratio is 108.5/140 = 0.775.
- The Proposed algorithms detect the first peak with probability of (1-0.968)/(1-0.95) ≈ 0.64. Their performance is near the optimal.
- Chen algorithm has the same probability above but the source must wait for the round-trip time to the longest destination (25 ms) until it receives feedback. This due to the delay of checking the switch congestion situation until taking the decision of sending BRM cell.


ACR of S1..Fahmy(0.05)

ACR of S1..Chen(0.05)

ACR of S1..Ros

ACR of S1..Proposed(0.05)

ACR of S1..Modified-Ros

ACR of S1..Modified-Proposed(0.05)

Figure 4.19: Allowed Cell Rate in Mbps for J.B. Config. with low threshold

- As the threshold is high, the possibility of sending more BRM Cells is increased. Therefore, the RM ratio is greater in Modified-Proposed graph than in Proposed algorithm.

The ratio of Modified-Proposed algorithm is approximately as the ratio of Chen Algorithm. The effect of the new RM ratio control scheme appears clearly in the transient period. In the Proposed algorithm graph, the ratio at 40 ms is ≈ 0.1 while in Modified-Proposed graph it is ≈ 0.19. The same can be shown with respect to Ros and Ros-Modified graphs.

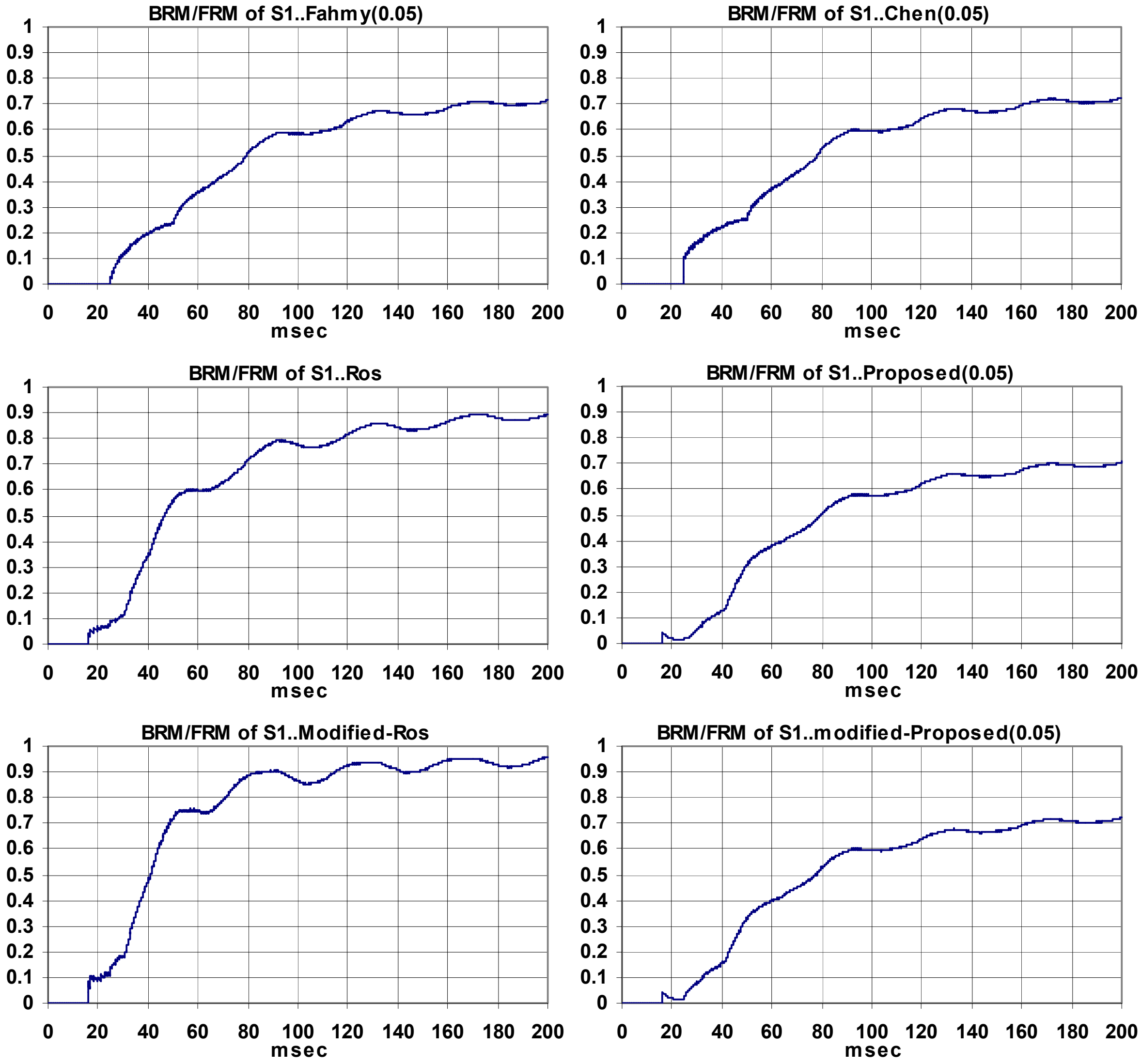


Figure 4.20: The RM ratio of the source for J.B. Config. with low threshold

Figures 4.19 and 4.20 illustrate the performance of the six algorithms with low threshold. In this case, the transient response of the proposed algorithms is faster than of Fahmy and Chen algorithms. Their performance is the nearest to Ros algorithm. By the periodical local congestion check beside the chance to send BRM cell if rate decrease is above the threshold, the proposed algorithms track the optimal solution from the second peak at ≈ 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.

Fahmy performance degrades to the "wait-for-all" performnce. SW2 waits 20 ms to collect information from all branches in absence of overload indication. The same behavior is performed by Chen algorithm in absence of local congestion check.

4.4.4 Comparison of the Algorithms

Table 4.2 shows a summary of the results of the comparison between the tested consolidation algorithms. Note that the main drawback of each algorithm is indicated in bold face.

Table 4.2: Comparison between Tested Algorithms

Algorithm	**Fahmy**	**Chen**	**Ros**	**Modified Ros**	**Proposed**
Implementation Complexity	>Medium	Medium	**High**	**High**	>Medium
Transient Response (low threshold)	**Slow**	**Slow**	Very fast	Very fast	Fast for overload
Transient Response (high threshold)	Very Fast for overload	Fast for overload	Very fast	Very fast	Very fast for overload
Consolidation Noise	Low	Low	Low	Low	Low
BRM:FRM at root	Lim =1	**May be >1**	Lim=1	Lim=1 but faster	Lim=1

In terms of complexity, our algorithm is more complex than both Fahmy and Chen, 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 thresholds offer reasonable fast response. However, Both Fahmy and Chen algorithms exhibit a slow transient response if the threshold is close to zero. While, our algorithm 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.

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 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 (lim in the table means the limit as time goes to infinity). 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. The new RM ratio control scheme led the ratio of Ros algorithm converges faster. Actually, the scheme effect is clear as the chance to send more BRM cells is increased. This may be achieved by a high threshold or a configuration with highly bottleneck rate of change.