Max-Min Rate Control Algorithm for Available Bit Rate Service in ATM Networks

Sudhakar Muddu
UCLA Computer Science Department
Los Angeles, CA 90095-1596
sudhakar@cs.ucla.edu

Christos Tryfonas*, Fabio M. Chiussi, and Vijay P. Kumar

AT&T Bell Laboratories
ATM Networks Research Department
101 Crawfords Corner Rd., Rm. 4D521
Holmdel, NJ 07733
fabio@big.att.com

Abstract

The definition of Available Bit Rate (ABR) service has been the focus of the recent activities of the ATM Forum. The Forum has adopted rate-based schemes as the standard for congestion control of ABR services, and has accepted *Enhanced Proportional Rate Control Algorithm* (EPRCA) as the recommended algorithm for the switch behavior. In this paper, we propose a new binary control algorithm, called *Max-Min Rate Control Algorithm* (MMRCA), which is fully compatible with the existing ATM standard. The MMRCA scheme uses minimum and maximum rate of all active connections to select which connections should be forced to decrease their rate during congestion, and uses additional congestion detection mechanism to prevent potential congestion by intelligently regulating selected connections. The MMRCA scheme converges to the same equal share for all connections as EPRCA, but achieves faster convergence time. The new scheme also requires smaller buffer sizes at the switches, and higher link utilization than the EPRCA scheme. The MMRCA scheme has significantly lower hardware complexity than EPRCA and other existing rate based schemes, since it eliminates the need for floating point division at the switch. The proposed scheme can be extended to implement Explicit Rate schemes with low hardware complexity.

*Currently at University of California, Santa Cruz, CA, 95060.

1 Introduction

Asynchronous Transfer Mode (ATM) has emerged as the technology of choice for broadband networks. Together with the conventional Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services supported in ATM Networks, a new service category called Available Bit Rate (ABR) has been recently introduced by the ATM Forum. Contrary to CBR and VBR, for which the service guarantees (e.g., delay, cell loss ratio, bandwidth) are negotiated at call set up time through admission control and bandwidth allocation, the ABR service guarantees a cell loss ratio only to those connections whose source dynamically adapts its traffic in accordance with feedback received from the network. The introduction of ABR service has been motivated by the need of sharing the available bandwidth among all active users, under traffic generated by highly bursty data applications. Most of these applications cannot predict their own traffic parameters at call set up time, thus an explicit guarantee of service at call set up time would be wasteful.

Since ABR service is inherently closed-loop, the congestion control scheme used to dynamically regulate the cell generation process of each connection by providing feedback information from the network is an integral part of ABR service. Congestion control for ABR service aims satisfying the following criteria: (i) maximal link utilization and rapid access to unused bandwidth, (ii) fair bandwidth allocation to each Virtual Connection (VC) and convergence to steady state, (iii) robustness against losses/errors in control information and against misbehaving users, and (iv) scalability with large number of VCs and network nodes. Although, the ABR standard does not require the cell transfer delay and cell loss ratio to be guaranteed or minimized, it is desirable for switches to minimize the delay and loss as much as possible.

The definition of the congestion control mechanism for ABR service has become the focus of the recent activities of the ATM Forum. Several mechanisms have been proposed [12, 8, 13, 14, 15, 17] and can be classified into two categories: *credit-based* and *rate-based*. The ATM Forum has adopted *rate-based* schemes as the standard for congestion control of ABR service. Rate based schemes, as the name implies, use feedback information from the network to control the rate at which each source can emit cells into the network. The control information is conveyed to the endpoints through special control cells called *Resource Management* (RM) cells, whose format has been defined by the ATM Forum. The Forum has also specified source and destination behaviors, while the behavior at the switches is left to the switch manufacturers.

The rate based schemes can be broadly divided into two categories: (i) *Binary Control schemes*, and (ii) *Explicit Rate schemes* (ER). Several binary control schemes have been proposed and the ATM Forum has accepted *Enhanced Proportional Rate Control Algorithm* (EPRCA) [9, 8] as the recommended algorithm for the switch behavior. In this paper, we propose a new binary control algorithm, called *Max-Min Rate Control Algorithm* (MMRCA), which is fully compatible with the existing ATM standard (i.e., uses the same RM cells, and source and destination behavior specified by the standard). The MMRCA scheme differs from the EPRCA scheme in that it uses minimum and maximum rate of all active connections to select which connections should be forced to decrease their rate during congestion. It also uses a congestion detection mechanism to quickly react to changing traffic conditions, and prevent potential congestion by intelligently regulating selected connections. The MMRCA scheme converges to the same equal share for all connections as EPRCA, but achieves faster convergence time. The new scheme also requires smaller buffer sizes at the switches, and higher link utilization than the EPRCA scheme. The MMRCA scheme has significantly lower hardware complexity than EPRCA and other existing rate based schemes, since it avoids the computation of the average rate for all connections, and eliminates the need for floating point division at the switch. The proposed scheme can be extended to implement ER schemes with low hardware complexity.

The rest of the paper is organized as follows. In Section 2, we review existing rate based congestion control schemes and in Section 3, we discuss the Enhanced Proportional Rate Control Algorithm (EPRCA), accepted by ATM Forum. In Section 4, we introduce the Max-Min Rate Control Algorithm (MMRCA). In Section 5, we present simulation results illustrating the performance characteristics of our scheme and the advantages over the EPRCA scheme. In Section 6 we present some preliminary considerations to build Explicit Rate schemes on top of MMRCA and other future extensions of this work. Finally, Section 7 offers some concluding remarks.

2 Rate-Based Control Schemes

In general rate-based approaches can be classified into two categories: negative feedback schemes and positive feedback schemes. In negative feedback schemes the feedback information is conveyed only when the network or the node falls into congestion. The negative feedback schemes are further divided, depending on the direction of the congestion notification from the congested node, into two approaches: Forward Explicit Congestion Notification (FECN) schemes [2, 4] and Backward Explicit Congestion Notification (BECN) schemes [3]. In FECN scheme, if an intermediate node (or switch) becomes congested, it will first convey information about the congested state to the destination, which will then notify the source of the congestion status. FECN schemes are based on end-to-end control, where the computation complexity of the congestion control algorithms resides mostly in the end systems and the intermediate switches are releaved of this complexity. Usually, the control information is conveyed in the forward direction using the Explicit Forward Congestion Indicator (EFCI) state in the payload type identifier (PTI) of the ATM cell.

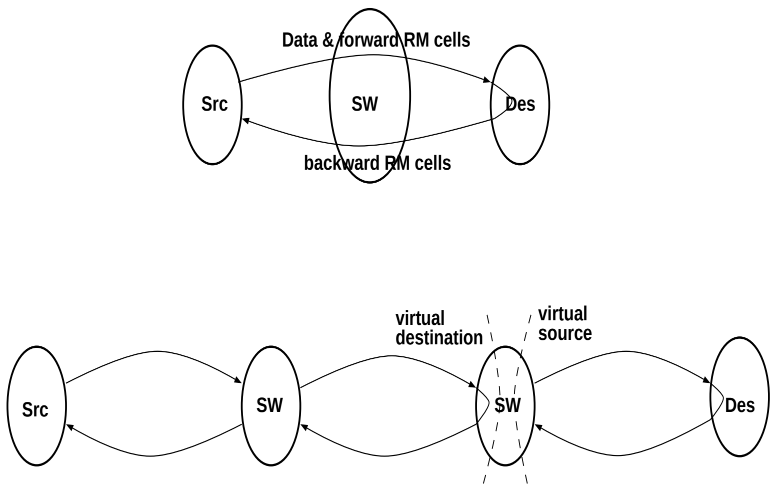


Figure 1: The top figure shown the flow of data and control (RM) cells. The bottom figure shows the segmentation of an intermediate node into virtual destination and virtual source.

BECN [3] uses similar mechanism to FECN except that the congestion notification is returned directly from the point of congestion to the source. Information about the state of the network, such as bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special control cells called Resource Management (RM) Cells. ATM Forum has adopted RM cell-based schemes to support congestion control under BECN mechanism. The source sends Forward RM cells periodically to the intermediate switches in the forward directions. During congestion, Backward RM cells are returned and sent back to source either from an intermediate switch or from the destination. In an extension of the scheme, any intermediate node can be divided into a virtual destination end system and virtual source end system as shown in Figure 1, and the RM cells can be sent back from the virtual destination nodes. BECN scheme requires more hardware in the switches than FECN scheme to detect and notify the source of congestion status. But BECN is more robust against faulty or noncompliant end systems because the network itself generates the congestion notification. Similar to FECN scheme, the polarity of the feedback information from the network is negative for BECN scheme. In both FECN and BECN schemes, if the congestion notification cells (RM cells) returning to the source experience extreme congestion and are dropped by the network, then the overall network may collapse [7] due to congestion buildup. The reason is that every VC will attempt to reach the peak cell rate and overload the queues in the absence of returned RM cells.

These potential catastrophic problems associated with negative feedback approaches led ATM Forum to develop a more robust scheme based on positive feedback mechanisms. The positive feedback schemes can be broadly classified into two categories:

- Binary Control Schemes [5, 6, 8], where the source rate is increased or decreased by a small amount (either a fixed quantity or a function of current source rate); and
- Explicit Rate Schemes [10, 13, 15], which compute explicitly the fair rate for each connection and then provide each source with the rate at which it should transmit cells.

The ATM Forum has defined RM cells to support both schemes, since RM cells can convey rate information in the form of either a single bit or a floating point number representing the exact rate.[1] In this paper we are mostly interested in Binary Control schemes since they are very attractive because of their simple implementation and reasonably good performance characteristics; Even though explicit rate schemes offer higher performance, they are considered expensive techniques with current hardware technology. Furthermore, the explicit rate schemes that have been proposed are built on top of binary control schemes.

We give a brief overview of some of the binary control schemes that are proposed in ATM Forum and the problems associated with such approaches. Recently Roberts et al [5] have proposed Proportional Rate Control Algorithm (PRCA), which eliminates the possibility of network collapse due to loss of control information. In the PRCA scheme a source only increases its cell

[1] Appendix A lists some relevant fields of RM cell.

rate when it receives an explicit positive indication from the destination or from any intermediate switch in the network. In the absence of positive indication the source will continually reduce its cell rate.[2] Also the destination and the switch nodes can send an explicit indication to decrease the source rate. This scheme is referred as proportional rate control algorithm because the increments and decrements in the cell rate of each connection are proportional to the current cell rate. However, the PRCA still suffers from problems such as the "beat down" problem [6] (also known as feedback starvation). Since each source of active connections that go through many congested links/switches has less opportunity to receive positive feedback than sources of connections that traverse fewer links/switches. In certain circumstances, once one of the feedback-starved sources decreases its cell rate to the minimum rate (MCR), it may likely remain at that rate indefinitely. Thus, fairness among connections cannot be achieved.

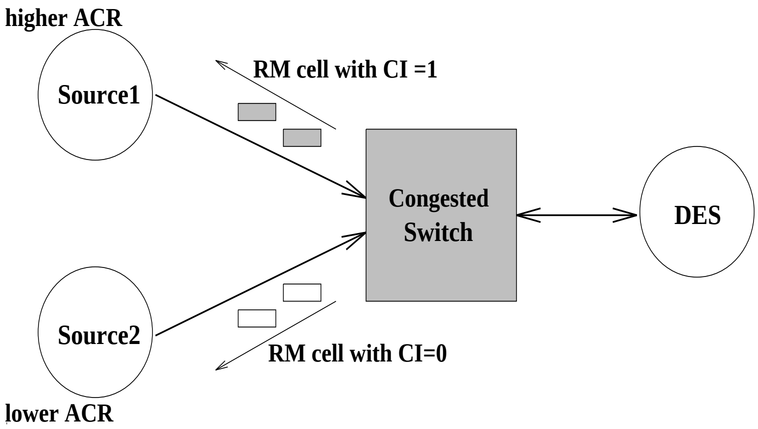


Figure 2: Intelligent marking at the switch depending on the source rates. CI (Congestion Indicator) is a field in RM cell to convey the congestion information (see Appendix A).

Two solutions that have been proposed for the "beat down" problem are: (i) *per-VC queuing*, and (ii) *intelligent (or selective) marking*. In *per-VC queuing* [8, 9] better fairness could be achieved because each connection, or group of connections, are maintained separately at the switch rather than using the same resources (buffers etc.) for all connections. This approach significantly increases the hardware complexity and memory requirements of the switches. *Intelligent marking* has the capability of sending congestion notification to particular sources rather than to all sources, and can achieve better fairness (see Figure 2). This overcomes the limitation of PRCA and other existing schemes which do not distinguish different VCs and reduce the rate of all VCs once congestion occurs. An improved version of PRCA, called *Enhanced Proportional*

[2] In the rest of the paper we refer to this continual decrease of source rate as periodic decrease of source rate.

Rate Control Algorithm (EPRCA), based on intelligent marking, has been proposed [8, 11], and adopted by the Forum. Section 3 below provides a complete description of the EPCRA approach.

3 Enhanced Proportional Rate Control Algorithm (EPRCA)

In this section we describe the EPRCA scheme, which is the recommended binary control scheme in the ATM Forum. EPRCA is an improved version of PRCA scheme to achieve better fairness among connections by sending congestion indication selectively to particular sources rather than to all sources.[3] The "beat down" problem was removed in EPRCA by selectively reducing the rate of source with large cell transmission rate. For implementing this mechanism, the EPRCA scheme maintains the mean of all connection rates ($MACR$) and then selectively reduces the source rates greater than the mean value. However, computation of the mean requires a floating point division, which increases the hardware complexity of the switch. Also the EPRCA scheme experiences large convergence times and buffer sizes, because there is only one threshold level (i.e., the value of the mean); thus, under congestion, all connections whose rate is above the mean are forced to decrease their rates equally, irrespective of their rates. In order to reduce the hardware complexity, it has been proposed to approximate the mean value using an exponential averaging technique [8], which requires only addition and shift operations. Since this approximation is far from accurate, and the performance of the intelligent marking approach depends very much on the value of the $MACR$ threshold value, the approximate scheme actually allocates bandwidth to the connections unfairly. We have observed that the performance results for the EPRCA scheme with approximate $MACR$ calculations are much worse than with exact $MACR$ calculations. nevertheless, in this paper we compare our scheme with the more effective EPRCA that uses exact $MACR$ computation. We now discuss in detail both the proposed switch behavior and the end system behavior under the EPRCA scheme.

3.1 Switch Behavior

As mentioned above, the control information is exchanged between end systems and the intermediate switches using a special cell called RM (Resource Management) cell. The source periodically

[3] Except for some modifications required to incorporate new features, EPRCA preserves backward compatibility with PRCA.

sends a forward RM cell every N_{RM} data cells. When the destination receives the forward RM cell, it returns the RM cell to the source as a backward RM cell with appropriate congestion information marked. RM cells contain a CI (Congestion Indication) bit that is used to carry congestion information to the source. The intermediate switches notify the source of congestion by marking the CI bit in the RM cells. If the CI bit is set then the intermediate switches are not allowed to reset CI bit as it could have been set by other nodes downstream. RM cells also contain a No Increase (NI) bit that is used to prevent the source from increasing its Allowed Cell Rate (ACR). It is typically used by intermediate switches when impending congestion is sensed. Also the switches can selectively inform some sources to increase their rate using the NI bit. If the NI bit is set the switches are not allowed to reset NI bit. EPRCA also uses the Current Cell rate (CCR) field in the RM cell to achieve a fair distribution of the available bandwidth, by selectively indicating congestion to sources with larger ACR values. The CCR field is set by the source to its current ACR when it generates the forward RM cell. The CCR field may not be modified by other network elements.

A switch which supports only PRCA is called an EFCI switch and switches supporting EPRCA with only intelligent marking are called Binary Enhanced Switches (BES). In BES switches, two threshold values on the queue length are used for indicating congestion: QT and DQT. When the queue length in the cell buffer of a BES switch exceeds QT, it is considered as congested and the switch performs intelligent marking. It selectively reduces the rate of all connections with ACR larger than the Mean ACR ($MACR$). The switch computes $MACR$ as:[4]

$$MACR = \frac{\sum_{i=1}^{N} CCR_i}{N} \tag{1}$$

where N is total number of active connections. A key issue in this scheme is the accurate calculations of $MACR$. Since the exact computation of the mean involves a floating point division[5] operation, the $MACR$ is typically approximated using an exponential Averaging factor (AV) as

$$MACR \approx MACR(1 - AV) + CCR * AV \tag{2}$$

where $AV = \frac{1}{16}$ [8].

[4] Another approach for selective marking is based on the *equal share* computation. The equal share for all the active connections is defined as the total available bandwidth over the total number of active connections. This scheme is similar to the above described EPRCA method except that equal share (EQ) is used as the threshold level instead of $MACR$ for intelligent marking.

[5] According to ATM Forum the rates are represented using 16 bit floating point representation [16].

When the queue length crosses the QT threshold, the switch marks the CI bit of the forward/backward RM cells if its CCR value exceeds $MACR * DPF$, where $DPF = \frac{7}{8}$ (Down Pressure Factor) for safe operation.[6] All other connections with rates (CCR) less than $MACR * DPF$ still continue to increase their rates. However, if the switch still remains congested and the queue length exceeds the DQT threshold, the switch is congested heavily; then, all connections have their rate reduced, irrespective of their CCR values.

3.2 Source and Destination End System Behavior

Since most of the computations and decisions are made at the switch, the source and destination behaviors are further simplified. The source periodically sends a forward RM cell every N_{RM} data cells.[7] When the destination receives the forward RM cell, it returns the RM cell to the source as a backward RM cell. During this process, the destination sets the CI bit of the backward RM cell either equal to the CI bit of the forward RM cell or equal to the EFCI status of the last incoming data cell , if any data cell has arrived after the forward RM cell. When a backward RM cell is received at the source with CI bit set (indicating congestion) then ACR is decreased by Additive Decrease Rate (ADR) as[8]

$$ACR = \text{MAX}\,(ACR - ADR, MCR) \tag{3}$$

To avoid collapse of the network due to conditions, such as loss of backward RM cells, the source end system continuously decreases its rate by Periodic Decrease Rate (PDR) at every data cell transmission time until it receives a backward RM cell, i.e.,

$$ACR = \text{MAX}\,(ACR - PDR, MCR) \tag{4}$$

Therefore, if a source receives an RM cell indicating rate increase, then the rate increase should first compensate the reduced rate since the source received the previous backward RM cell ($N_{RM} * PDR$) and then increase the rate by Additive Increase Rate (AIR) as

$$ACR = \text{MIN}\,(ACR + N_{RM} * PDR + AIR, PCR) \tag{5}$$

[6] The DPF factor is defined to include those VCs whose rate is very close to $MACR$, which should also be forced to reduce their rate during congestion.

[7] Since RM cells are sent periodically after N_{RM} data cells instead of after a fixed period EPRCA scheme is also referred as Counter-Based approach.

[8] Even though we discuss additive decrease, it could be modified to multiplicative decrease of ACR.

where Peak Cell rate (PCR) is the maximum rate at which the source can transmit cells. ADR and AIR need not be fixed quantities, but could be functions of the current cell rate of the sources. Even under heavy congestion with all RM cells being discarded, the network does not collapse because the sources always decrease their rate whenever they source do not receive an RM cell.

3.3 Enhancements to Source End System Behavior

A modification to the periodic decrease of the source rate has been recently accepted by the Forum. In the scheme described above, having the source continuously decrease their rate immediately after receiving the backward RM cells, makes them transmit at a lower rate even if there is no congestion, and RM cells keep arriving without fail, thus affecting link utilization. Thus, this problem can be reduced by postponing the source rate decrease to some predefined time after the forward RM cell transmission. The Forum has selected to decrease the source rate only after $TOF * N_{RM}$ data cells are transmitted, following the transmission of the forward RM cell, if no backward RM cell is received during during that period (for example, $TOF = 2$).

In addition to this modification, we also propose to decrease the source rate after every N_P data cells are transmitted, where the range of N_P is 1 to N_{RM}, instead of decreasing the rate after every data cell. In this case, the source rate is increased when the source receives a backward RM cell indicating a rate increase, but the amount of increase depends on the arrival time of the backward RM cell. If the RM cell occurs before transmitting $TOF * N_{RM}$ data cells after the forward RM cell transmission, then the rate is increased by AIR only, i.e.,

$$ACR = \mathrm{MIN}\,(ACR + AIR, PCR) \tag{6}$$

If the backward RM cell arrives after transmitting K ($K > TOF * N_{RM}$) data cells, then the rate increase should compensate the total periodical decrease rate in the period $K - TOF * N_{RM}$ cells (rate reduced by PDR at every N_P cells) and also increase the rate additively by AIR, i.e.,

$$ACR = \mathrm{MIN}\left(ACR + \frac{(K - TOF * N_{RM})}{N_P} * PDR + AIR, PCR\right) \tag{7}$$

We have observed that postponing the periodical source rate decrease to $TOF * N_{RM}$ data cells after forward RM cell transmission and decreasing the rate after every N_P data cells, provides more flexibility for the congestion control algorithms.

4 Max-Min Rate Control Algorithm (MMRCA)

The above described EPRCA scheme has the following drawbacks:

1. EPRCA uses the mean of all VC rates as the threshold for selectively controlling the rate of the connections. If the $MACR$ is computed *exactly* then a floating point division unit is needed at each of the switch nodes, which dramatically increases the hardware complexity; (a 16-bit floating point divider requires a very significant silicon area).
2. even with exact $MACR$ computation the EPRCA scheme has unfairness and convergence problems. Since there is only one threshold level ($MACR$), during congestion, all connections whose rate is above the mean are forced to decrease their rates equally, irrespective of their rates.
3. the approximation of the mean using the exponential averaging technique to avoid division operation is not accurate, and leads to substantial additional unfairness in the bandwidth allocation between the connections.

In this section we propose a new binary congestion control scheme, called Max-Min Rate Control Algorithm (MMRCA), which offers higher performance than EPRCA, and it is easier to implement at the switch. This new scheme is fully compatible with the existing ATM Forum standard, since it uses the RM cell as specified by the standard. The MMRCA scheme differs from the EPRCA scheme in that it uses minimum and maximum rate of all active connections to select which connections should be forced to decrease their rate during congestion. It also uses a congestion detection mechanism to quickly react to changing traffic conditions, and prevent potential congestion by intelligently regulating selected connections. The MMRCA scheme performs better than EPRCA because it has more levels of selective threshold, thus achieving more flexibility is selectively controlling the source rates. The MMRCA scheme converges to the same equal share for all connections as EPRCA, but achieves faster convergence time. [9] The new scheme also requires smaller buffer sizes at the switches, and higher link utilization than the EPRCA scheme. The MMRCA scheme has significantly lower hardware complexity than

[9] The convergence time, refers to the time difference between the start of the connections to the point when all connections get equal share of the available bandwidth.

EPRCA, since it only keeps track of the minimum and maximum rate of all connections and avoids the computation of the average rate for all connections, thus eliminating the need for floating point division at the switch.

In our scheme all the connections are divided into three different sets: (i) set of connection with rates around the maximum rate, (ii) set of connection with rates between maximum and minimum rate, and (iii) set of connection with rates around the minimum rates. During congestion and depending on the level of congestion, instead of reducing the rate of all connections only the connections which are more probable to aggregate congestion are forced to reduce their rates. In the following, we propose our basic scheme, referred to as MMRCA_Basic, which only uses selective marking based on maximum and minimum rate, and two enhancements the basic scheme, called MMRCA with Rate of Change of Queue length (MMRCA_RQL) and MMRCA with Total Rate (MMRCA_TR), which also use partial congestion detection mechanisms and depend upon different thresholds to selectively control the connection rates.

4.1 Description of the Algorithm

The MMRCA_Basic scheme uses only queue length thresholds for congestion detection. When the queue length exceeds the threshold QT, the switch selectively indicates congestion to all the connections that have rates higher than the *minimum* source rate, MIN. All connections having a current rate CCR higher than $MIN * IPF$ are indicated to reduce their rate. The Increase Pressure Factor (IPF) is similar to the DPF factor used in EPRCA scheme for safe operation to include the connections whose rates are very close the minimum rate VC; we use $IPF = 9/8$. The connections with rates below $MIN * IPF$ are still allowed to increase the rate if the difference between maximum rate and minimum rate is not too small, i.e., $MAX - MIN \geq MX\_MN\_DIFF$. However, if the difference between the maximum and minimum rates is small, then the connections with CCR below $MIN * IPF$ are also indicated to reduce their rates.[10] The intuition behind this argument is that, if the maximum and minimum of all active connections are close to each other, then it is not fair to selectively decrease some connections; in this case, all connections (including maximum and minimum rate connections) should be subject to the same congestion control strategy. The maximum and minimum rates and the corresponding VC

[10] It turns out that the MMCRA schemes are not sensitive to the difference between maximum rate and minimum rate (MX_MN_DIFF) and we have tried various values between 1% to 10%.

numbers are updated only when an RM cell arrives from any active connection, as shown in the pseudo-code given in the Figure 3. Even with the intelligent marking and selective reduction of rates the switch may still remain congested. Therefore, if the switch becomes *very* congested, as indicated by the queue length exceeding a higher threshold DQT, the rates of all connections is reduced. The pseudo-code of the MMRCA_Basic algorithm at the switch is given in Figure 4.

```
/* Update of maximum id and rate */
  if connection i = MAX_VC /* RM cell belongs MAX_VC */
  then update MAX rate
  else
      if CCR_i > MAX /* Current VC is the MAX_VC */
      then MAX_VC = i; MAX = CCR
/* Update of minimum id and rate */
  if connection i = MIN_VC
  then update MIN rate
  else
      if CCR_i ≤ MIN
      then MIN_VC = i; MIN = CCR
```

Figure 3: The pseudo-code for the update of maximum and minimum rates and id numbers. The code is executed only upon the arrival of RM cell from any active connection.

We have formulated two enhancements of the basic scheme by providing support for detection of partial congestion and by using separate selective mechanism for marking the connections during partial congestion. The first enhancement, MMRCA_RQL, uses rate of change of queue length to provide detection of partial congestion. The switch is said to be in partial congestion if the queue length increases by a fixed amount (RQL) in a fixed time interval (N_{QL} cell times). Even though the rate of change of queue length does not correspond to any congestion of resources, we have observed that it is an useful indication of partial congestion. If partial congestion is detected, more severe congestion can be avoided by selectively reducing the rate of sources with large ACR values. If the switch is partially congested, all connections that have rates around the *maximum* source rate will be decreased. More specifically, the switch reduces the rate (i.e., marks the CI bit in RM cell) for a connection if its CCR value exceeds $MAX * DPF$, where DPF (Down Pressure Factor) is 7/8. As in the EPRCA scheme (and in the ATM standard) the DPF factor is used for safe operation to include those connections whose rates are very close to the maximum rate. All other connections with rates CCR less than $MAX * DPF$ still continue

Algorithm MMRCA_Basic
Input: RM Cell from Connection i
Output: Marking of RM cell
MAX_VC: Current maximum rate connection MIN_VC: Current minimum rate connection MAX: Current maximum rate MIN: Current minimum rate

```
if Queue Length ≥ DQT /* Highly Congested Queue */
then decrease rate of connection i
else
    if Queue Length ≥ QT /* Just About Congested Queue */
    then
        if CCR_i ≥ MIN * IPF
        then decrease rate of connection i
        else
            if MAX − MIN ≥ MX_MN_DIFF
            then
                if connection i ≠ MAX_VC connection
                then increase rate of connection i
                else decrease rate of connection i
            else decrease rate of connection i
    else
            if Uncongested /*Queue is Uncongested */
            then
                if connection i ≠ MAX_VC connection
                then increase rate of connection i
                else do not touch connection i
```

Figure 4: The pseudo-code for the MMRCA_Basic algorithm. It uses only thresholds on queue length for congestion detection and does not support partial congestion detection.

to increase their rate. The pseudo-code of the MMRCA_RQL algorithm is given in Figure 5.

The second enhancement, MMRCA_TR, is similar to the MMRCA_RQL scheme, but uses the total rate of all active connections as the method for partial congestion detection. If the total rate of all incoming active connections is greater than the outgoing link rate, then the switch is said to be under partial congestion. Under such conditions, connections whose rate is close to the maximum rate are forced to reduce their rates. The pseudo-code of the algorithm is given in the Figure 6.

```
Algorithm MMRCA_RQL
Input: RM Cell from Connection i
Output: Marking of RM cell
Output: Marking of RM cell
  MAX_VC: Current maximum rate connection
  MIN_VC: Current minimum rate connection
  MAX: Current maximum rate
  MIN: Current minimum rate
  if Queue Length ≥ DQT /* Highly Congested Queue */
  then decrease rate of connection i
  else
     if Queue Length ≥ QT /* Just About Congested Queue */
     then
        if CCR_i ≥ MIN * IPF
        then decrease rate of connection i
        else
           if MAX − MIN ≥ MX_MN_DIFF
           then
              if connection i ≠ MAX_VC connection
              then increase rate of connection i
              else decrease rate of connection i
           else decrease rate of connection i
     else
        /* Partial (or potential) Congestion */
        if Increase in Queue Length in N_QL time units ≥ RQL
        then
           if CCR_i ≥ MAX * DPF
           then
              if connection i ≠ MIN_VC connection
              then decrease rate of connection i
              else increase rate of connection i
           else increase rate of connection i
        else
           if Uncongested /*Queue is Uncongested */
           then
              if connection i ≠ MAX_VC connection
              then increase rate of connection i
              else do not touch connection i
```

Figure 5: The pseudo-code for the MMRCA_RQL algorithm, where the partial congestion detection is done using rate of change of queue length.

4.2 Fairness and Convergence Time

When a new connection is established and the current available bandwidth is fairly allocated, then **MMRCA** converges to the equal share for all the active connections in finite time. The

Algorithm MMRCA_TR
Input: RM Cell from Connection i
Output: Marking of RM cell
Output: Marking of RM cell

```
  MAX_VC: Current maximum rate connection
  MIN_VC: Current minimum rate connection
  MAX: Current maximum rate
  MIN: Current minimum rate

if Queue Length ≥ DQT /* Highly Congested Queue */
then decrease rate of connection i
else
    if Queue Length ≥ QT /* Just About Congested Queue */
    then
        if CCR_i ≥ MIN * IPF
        then decrease rate of connection i
        else
            if MAX − MIN ≥ MX_MN_DIFF
            then
                if connection i ≠ MAX_VC connection
                then increase rate of connection i
                else decrease rate of connection i
            else decrease rate of connection i
    else
        /* Partial (or potential) Congestion */
        if Total Input Rate ≥ Output Link Rate
        then
            if CCR_i ≥ MAX * DPF
            then
                if connection i ≠ MIN_VC connection
                then decrease rate of connection i
                else increase rate of connection i
            else increase rate of connection i
        else
            if Uncongested /*Queue is Uncongested */
            then
                if connection i ≠ MAX_VC connection
                then increase rate of connection i
                else do not touch connection i
```

Figure 6: The pseudo-code for the MMRCA_TR algorithm, where the partial congestion detection is achieved by comparing the total input rate with output link rate.

following lemma is used to prove the fairness.

Lemma 1: If a new connection i is established with an initial rate r_i at time t_1 and the other existing connections are transmitting at the current equal share $EQ(t_1)$, then all connections (including the new connection) converge to a fair bandwidth allocation of *equal share* $(EQ(t_2))$ at time $t_2 > t_1$.

From the above lemma, if there are $N-1$ active connections already in the network and a new connection established, then the connection will reach a steady state equal to the total available bandwidth (ABW) over all active connections at time t_2, i.e.,

$$EQ_1(t_1) = \frac{ABW}{N-1} \qquad EQ_2(t_2) = \frac{ABW}{N}$$

The convergence time for this new connection is defined as $(t_2 - t_1)$. We have computed an upper bound for the convergence time. When a new connection is established, it starts with an initial rate of ICR and increases its rate when it receives a backward RM cell from the switch with no congestion indication. Assuming backward RM cells are not lost, the source rate is increased additively by AIR (see Equation 6). We also assume that D is an upper bound on the propagation delay between the source and the switch. Since the forward RM cells are sent after every N_{RM} data cells, the backward RM cells, and hence the steps of increase in source rate, occur every N_{RM} data cell. Before the new connection is established, we assume that all the active connections have reached a stable value of bandwidth EQ_1. The number of steps (S) required for the new connection to reach a new stable value EQ_2 is

$$S = \left| \frac{EQ_2 - ICR}{AIR} \right|$$

Each step consists of N_{RM} data cells and one RM cell and time taken for each step depends on the rate of transmission. The first step period is equal to $2D$ because the first backward RM cell arrives after round trip propagation delay. Using the rate of transmission in the second period as $ICR + AIR$ the time taken for transmitting $N_{RM}+1$ cells is $\frac{(N_{RM}+1)}{ICR+AIR}$. Therefore, the convergence time for the new connection to reach the *equal share* is given by

$$\begin{aligned}
T_{inc} &= 2D + \frac{N_{RM}+1}{ICR+AIR} + \frac{N_{RM}+1}{ICR+2*AIR} + \ldots + \frac{N_{RM}+1}{ICR+(S-1)*AIR} \\
&= 2D + \frac{(N_{RM}+1)}{AIR}\left[\sum_{k=1}^{S-1} \frac{1}{ICR/AIR+k}\right] \\
&= 2D + \frac{(N_{RM}+1)}{AIR}\left[\sum_{k=1}^{ICR/AIR+S-1} \frac{1}{k} - \sum_{k=1}^{ICR/AIR} \frac{1}{k}\right]
\end{aligned}$$

The sum of finite harmonic series can be upper bounded as [1]

$$\sum_{k=1}^{N} \frac{1}{k} < C + \ln(N) + \frac{1}{2N} \tag{8}$$

where $C = 0.577$. Assuming $\frac{ICR}{AIR}$ is an integer and substituting the above upper bound of the harmonic series sum in the convergence time, we get

$$\begin{aligned} T_{inc} &< 2D + \frac{(N_{RM}+1)}{AIR}\left[\ln\left(\frac{ICR/AIR + S - 1}{ICR/AIR}\right) + \frac{1}{2*(ICR/AIR + S - 1)} - \frac{1}{2*(ICR/AIR)}\right] \\ &= 2D + \frac{(N_{RM}+1)}{AIR}\left[\ln\left(\frac{EQ_2 - AIR}{ICR}\right) + \frac{AIR}{2}\left(\frac{1}{EQ_2 - AIR} - \frac{1}{ICR}\right)\right] \end{aligned}$$

If there are N active connections, then the equal share is proportional to $\frac{1}{N}$. Therefore, the algorithm will converge to a stable allocation within

$$\begin{aligned} T_{inc} &= O\left(2D + \frac{1}{AIR} + \frac{1}{EQ_2 - AIR}\right) \\ &= O\left(2D + \frac{\ln(N)}{AIR} + N\right) \end{aligned} \tag{9}$$

Similarly, the convergence time for the existing connections, which are at a stable value of EQ_1, to reach the new *equal share* EQ_2 can be obtained as

$$\begin{aligned} T_{dec} &= O\left(2D + \frac{1}{ADR} + \frac{1}{EQ_1 - ADR}\right) \\ &= O\left(2D + \frac{\ln(N)}{ADR} + N\right) \end{aligned} \tag{10}$$

The total convergence time can be obtained by choosing the maximum of Equation (9) and Equation (10), i.e., $T_{cov} = \text{MAX}(T_{inc}, T_{dec})$

5 Simulation Results

In this section we present simulation results to illustrate the performance of the MMRCA scheme for ABR traffic in different scenarios of ATM networks, and to qualify the advantages in terms of convergence time, buffer size, fairness, and utilization over the EPRCA scheme. The network topology that we have used in our simulations is shown in Figure 7. The network consists of four sources, with one VC per source, connected via four different links to an ATM switch. The switch is assumed to be non-blocking, and output buffered with FIFO scheduling. The switch is connected with a single output link to a destination. For the same network topology, we use different link propagation delays to study the impact of our algorithm. We also assume the sources are greedy (or persistent) and can transmit cells at the Allowed Cell Rate (ACR). Although in practice most sources are not persistent, the fairness and convergence characteristics of a scheme can be better illustrated using greedy sources rather than using non-greedy sources. We consider zero cell loss for all the schemes simulated. In the simulation, we have normalized the absolute time by considering the transmission time of one ATM cell as the time unit (or cell time unit). In the case of a link speed equal to 155Mbps, the cell time unit for transmitting one ATM cell of 53bytes is about $2.75\mu s$.

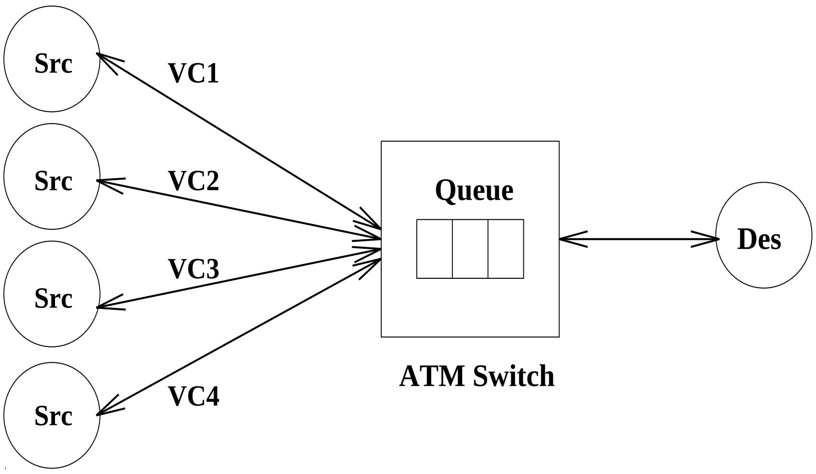


Figure 7: A simple local area network (LAN) topology with one switch and four VCs. All the four links are of the same length and have the same propagation delays.

The various simulation parameters used in this study for both MMRCA and EPRCA scheme are listed in Table 1: we have chosen similar set of of switch and end system parameters as in previous schemes [13, 19]. Except for the MX_MN_DIFF parameter, all the other parameters are used by both EPRCA and MMRCA schemes. In practice, these parameters may be chosen for each connection as a function of the network distance and can be specified during the connection setup time for each VC. In the different scenarios that we discuss in this paper, we have considered different lengths and propagation delays for the links of the network.

With both EPRCA scheme and MMRCA schemes, all the active VCs receive equal share (EQ) of the available bandwidth once the network is in steady state. The performance of the algorithm is determined by the convergence time, link utilization, and the buffer space required in the switch. For the simulation of EPRCA scheme we use *exact* mean computation. However,

Parameter	Values
PCR	100.0%
MCR	1.0%
ICR	5.0%
ADR	1.0%
PDR	0.1%
AIR	1%
MX_MN_DIFF	10.0%
DPF	7/8
IPF	9/8
N_{RM}	32
QT	30
DQT	60
N_{QL}	30
RQL	2

Table 1: The simulation parameters used for both EPRCA and MMRCA schemes. The percentage in the above parameters refer to the percentage of the total available bandwidth on the link.

in practice, in order to reduce hardware complexity, the mean computation is approximated (see Section 3) and the resulting convergence time, and link utilization of EPRCA are substantially than those obtained using exact mean computation. Even with exact mean computation, the performance results for the EPRCA scheme are inferior to the MMRCA approaches. Even the simple MMRCA_Basic scheme performs better than EPRCA. Furthermore, we show that, both MMRCA_RQL and MMRCA_TR algorithms have better convergence time and less buffer size over the MMRCA_Basic scheme because these two algorithms have support for partial congestion detection and have an extra threshold level for selective control of connection rates. MMRCA_TR approach performs better than MMRCA_RQL, but requires more hardware complexity because it requires the computation of the total sum of incoming rates. Hence, MMRCA_RQL and MMRCA_TR methods offer a tradeoff between hardware complexity and performance. In any case, both these schemes require substantially less hardware complexity than the EPRCA scheme.

In the following, we compare the performance characteristics of our MMRCA scheme with EPRCA for Local Area Network (LAN) and for Wide Area Network (WAN) configurations. In the case of LAN, we consider two scenarios: (i) Homogeneous Non-bottlenecked connections (ii)

Homogeneous Bottlenecked Connections. Homogeneous connections refers to connections that use links with the same propagation delay. A connection is said to be bottlenecked when the rate of the connection cannot be increased beyond a certain limit due to certain network conditions or to lack of network resources. In the steady state all connections converge to equal share (EQ), defined as the total available rate over number of active connections; if for some reason a connection cannot reach the equal share value in steady state, then it is also considered a bottlenecked connection. Usually, a connection spans over multiple nodes, and could get bottlenecked at any node in the network. In our simulation, since we consider a simple topology with only one switch, and four different links, we make one connection bottleneck by forcing that connection PCR (peak cell rate) to be equal to Initial Cell Rate (ICR).

5.1 Local Area Network (LAN)

In LAN configuration, both EPRCA and MMRCA schemes are studied for the simple network of Figure 7 with four homogeneous links (i.e., with identical length and propagation delays). The length of each link is 22 miles and has a propagation delay of $5\mu s/mile$. In terms of normalized time, the propagation delay of all links is 40 cell time units at 155 Mbps link speed.

5.1.1 Homogeneous Non-bottlenecked Network Model

In this case, we consider all the four VCs to be homogeneous and non-bottlenecked. Ideally, if no connections are bottlenecked then all VCs should reach a final rate equal to equal share. We first consider the case when VC1 and VC2 start their connection at time $t = 0$ and then the other two VCs join at time $t = 20000$ cell units. The instantaneous bandwidth of each VC and the total link utilization for the EPRCA scheme are plotted in Figure 8a. The maximum buffer size required for this configuration is 150 cells and the average link utilization for this configuration is 91.4%. Using the MMRCA_Basic scheme, the convergence of all VCs to equal share is faster and has less oscillations (See Figure 8b); also the maximum buffer size improves to 83 cells and the utilization increases to 91.6%. The performance results are even better for MMRCA_RQL and MMRCA_TR algorithms, as shown in Figures 8c and 8d. The maximum buffer size required are 66 and 41, respectively. Clearly, in this case of a small network topology with four links, the MMRCA schemes converge much faster than the EPRCA method.

We have seen that the MMRCA schemes perform better than EPRCA for large number of connections. We have considered the same network topology as of the previous case, but with 20 incoming VCs to the switch. We consider the case when two VCs (VC1 and VC2) start their connection at time $t = 0$; then after every 20000 cell time units two VCs join until all 20 VCs join. The total simulation time is 250000 cell times. The instantaneous bandwidth of each VC and the total link utilization for the EPRCA scheme are plotted in Figure 9a. The average buffer size required is 39, the maximum buffer size is 168 and the average link utilization for this configuration is 96.0%. With MMRCA_Basic scheme, we observe slightly larger oscillations, i.e., lower link utilization (see Figure 9b), but better average and maximum buffer size than the EPRCA scheme. With MMRCA_RQL and MMRCA_TR approaches as shown in Figures 9c and 9d, we observe significantly less osscillations and faster convergence time than EPRCA (in Figures 9c, $RQL = 2$). Also the size of the average and maximum buffer required are considerably smaller than with EPRCA. Table 2 summarizes and compares various performance characteristics of all these schemes. We have also simulated the MMRCA_RQL scheme with rate of change of queue length parameter $RQL = 0$ and the corresponding bandwidth allocation is shown in Figure 10. The average and maximum buffer size for this case are 13 and 107, and the link utilization is 93.2%. Comparing these results with those obtained for $RQL = 2$, we observe that buffer size is improved and link utilization is reduced. Therefore, the MMRCA_RQL scheme can be optimized for different scenarios using parameters, such as RQL and N_{QL} to obtain good performance results.

5.1.2 Homogeneous Bottlenecked Network Model

We now consider a special case of homogeneous LAN configuration where one of the VCs is bottlenecked. Ideally, the unused bandwidth of the bottlenecked connection should be equally divided among the remaining non-bottlenecked connections. We simulate the same network topology shown in Figure 7. The connections VC1 and VC2 start at time $t = 0$ and the other two VCs join at time $t = 20000$ cell time units. The total simulation time is 70000 cell time units. We assume that the VC1 connection cannot exceed the transmission rate equal to the initial cell rate (ICR), which is 5% of the link bandwidth. The instantaneous bandwidth and the total link utilization for the EPRCA scheme are plotted in Figure 11a. Quite interestingly, the bandwidth allocated to non-bottlenecked VCs does not converge, and some VCs are starved. This unfairness or VC starvation is similar to the "beat down" experienced for PRCA scheme. The reason for this unfairness is that the computation of the mean cell rate includes all active connections. This mean ($MACR$) value is always less than the non-bottlenecked equal share, which is equal to total non-bottlenecked bandwidth over non-bottlenecked connections. Therefore, those connections which start from rate less than the mean value will never reach the non-bottlenecked equal share, and hence are unfairly discriminated over the connections which have a higher rate. This unfairness can be resolved in the case of EPRCA scheme by separately marking each connection as bottlenecked or non-bottlenecked and then computing the mean (or average) over only non-bottlenecked connections. Obviously, this mechanism requires extra hardware to keep track of bottlenecked connections. Another disadvantage of the EPRCA scheme is, that the bottlenecked connection could at times be forced to decrease its already lower rate during congestion. This can be observed in the instantaneous bandwidth allocation of bottlenecked VC1 connection in Figure 11a, where VC1 bandwidth at times goes below its initial cell rate (ICR). This is clearly unfair to VC1, since other connections are still allowed to transmit at much higher rates.

We have simulated the homogeneous LAN bottlenecked configuration using MMRCA_Basic and the two enhancement schemes MMRCA_RQL, and MMRCA_TR. Contrart to the EPRCA scheme, we observe that all three methods have no fairness problem. and all converge to equal share of the available bandwidth as shown in Figures 11b, 11c, and 11d. As expected, the MMRCA_TR scheme has faster convergence time and smaller buffer size. Table 2 provides a comparison of the performance results for all these algorithms. An other advantage of the MMRCA schemes is that, it achieves convergence without using any extra hardware mechanism to keep track of bottlenecked connections, which is required with the EPRCA scheme to achieve equal share in the steady state.

5.2 Wide Area Network (WAN) Model

Finally, we have also tested MMRCA schemes for a wide area network (WAN) configuration with links of different propagation delays as shown in Figure 12. We consider a network configuration with four VCs, where one connection (VC1) is longer and has higher propagation delay than to the other connections. The length of the VC1 link is 220 miles and the propaga-

Network Model	EPRCA Scheme			MMRCA Scheme Algorithm 2			MMRCA Scheme Algorithm 3		
	Conv Time	**Buffer Size Avg/Max**	**Avg Util**	**Conv Time**	**Buffer Size Avg/Max**	**Avg Util**	**Conv Time**	**Buffer Size Avg/Max**	**Avg Util**
Non-bottlenecked LAN (N=4)	19000	33/150	91.4%	8000	19/66	91.6%	5000	10/41	93.2%
Non-bottlenecked LAN (N=20)	25000	39/168	96.0%	7000	24/123	95.5%	4000	18/98	96.5%
Bottlenecked LAN (N=4)	-	30/103	89.5%	12000	15/71	90.0%	6000	1/22	89.0%
Non-bottlenecked WAN (N=4)	40000	43/153	88.4%	9000	18/94	89.0%	9000	13/130	90.0%

Table 2: A comparison of EPRCA and MMRCA schemes using performance metrics such as convergence time, average/maximum buffer size, and average output link utilization. Here, the convergence time refers to the time difference between when VC2, VC3 starts to the point where all VCs share the bandwidth equally. The convergence time is expressed in terms of cell time units .

tion delay in one direction is 400 cell time units or 1.1 msecs. The other links have the same length (22 miles), and a propagation delay of 40 cell time units. We assume that VC1 and VC2 start the connection at time $t = 0$ and then the other two VCs join at time $t = 20000$. Again, the total simulation time is 70000 cell times. The instantaneous bandwidth and the total link utilization for the EPRCA scheme are plotted in Figure 13a. Because of the long propagation delay of VC1 the bandwidth of all VCs does not converge properly and the network experiences wide oscillations. In contrast, with the MMRCA schemes, all the VCs converge very well to the equal share and the average link untilization is higher, as shown in Figures 13b, 13c, and 13d. Also the average and maximum buffer size required with the MMRCA schemes are much smaller than for EPRCA scheme. A comparison of different performance metrics for EPRCA scheme and MMRCA schemes is given in Table 2.

6 Support for Explicit Rate Mechanism in MMRCA

The MMRCA schemes proposed above are algorithms for one-bit congestion control algorithms. Even though these algorithms solve the "beat down" problem and provide selective marking, these schemes require large buffer sizes as the the propagation delays increase and the number of VCs increase. Recently, to solve this problem an *Explicit rate mechanism* [8, 13, 10] is proposed. Under this scheme the rate of each connection is explicitly reduced/increased to a pre-computed value instead of reducing/increasing by a fixed quantity. That is, the switch will have the responsibility for determining the cell transmission rates of all the connections. A separate field called ER field is provided in the RM cells for communicating the explicit rate value computed in the switch to the end systems. A switch implementing explicit rate setting is called *Explicit Down Switches* (EDS). The explicit rate algorithms proposed [8, 13, 10] provide support for both intelligent marking and explicit rate setting mechanism. These schemes maintain mean of all connections ($MACR$) for intelligent marking (i.e. are based on the EPRCA scheme) and selectively control the source rates by setting the ER field of backward RM cell. Certainly, the complexity of the switches providing explicit rate is much higher; but it helps to reduce the convergence time and the buffer space required.

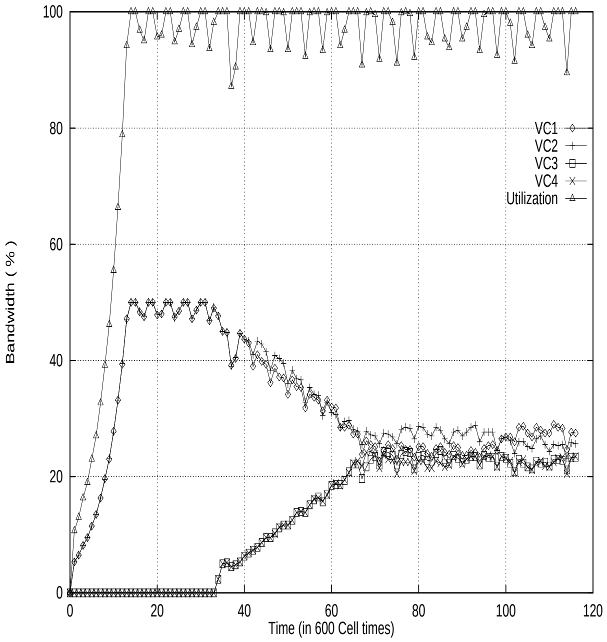
The aim of MMRCA schemes was to provide intelligent/selective marking and fairly allocate the available bandwidth to all the active connection with less switch complexity. We believe that our MMRCA schemes can be modified to support explicit rate mechanism. One simple way to implement the explicit rate is to keep track of available bandwidth and the fair share for each connection can be obtained by dividing available bandwidth over number of active connections. Also using the selective marking only those connections which exceed the fair share by large amount will be decreased first and then penalizing those connections which are just above the fair share etc. However, this method requires as much complexity as the previous schemes [8, 13, 10]. But we believe this scheme will perform better because it is based on MMRCA scheme fop congestion detection and selective marking, where as the previous schemes are based on EPRCA methodology.

Another possible implementation for explicit rate support on MMRCA scheme is to use a small state machine for each connection at the switch. Depending on the present congestion type at the switch (low, high, partial etc.) each connection changes state in its state machine. Each state is associated with pre-computed (or determined during call setup phase) explicit rate value and this explicit rate is communicated the end systems through $ERfield$. The state machine can be easily implemented; for example, we need only 4 bits per connection to implement a state machine with 16 states. We believe that combining MMRCA scheme with the concept of state machine mechanism for explicit rate support will improve the performance and reduces the hardware complexity [18].

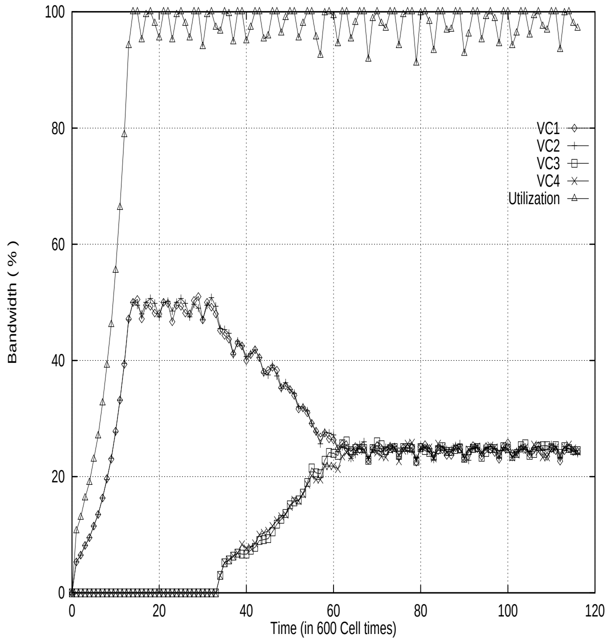
7 Conclusion and Future Work

In this paper, we have proposed a new binary congestion control algorithm for ABR called Max-Min Rate Control Algorithm (MMRCA), and showed that this scheme offers better performance than the EPRCA in terms of convergence, buffer size, and utilization. Also the hardware complexity is significantly reduced, since this scheme only needs to keep track of minimum and maximum rate of all connections. EPRCA, similarly to other existing schemes, uses either exact mean computation for its selection methodology, thus requiring large hardware complexity, or approximate techniques, thus leading to poor performance. In any case, MMRCA outperforms even EPRCA with exact computation of the mean rate.

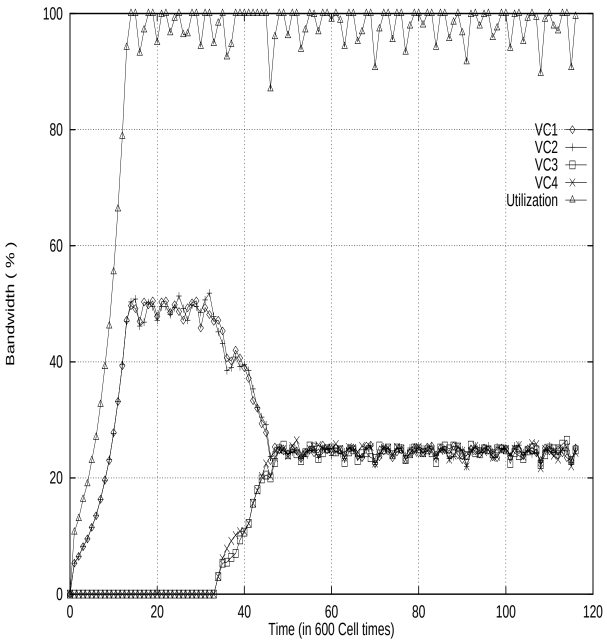
We have also proposed two enhancements of the basic MMRCA scheme, which use either the rate of change of queue length or the total rate of all incoming connections for detection of partial congestion. Our results indicate that our MMRCA algorithms can be optimized using the rate of change of queue length parameters to yield even better performance results. The algorithm could be further modified to use both conditions together or separately for different types of partial congestion detection. We intend to modify the algorithm to make use of another threshold level based on sum of minimum rate and maximum rate of all active connections ($\frac{MIN+MAX}{2}$), for intelligent selection methodology. Finally, the MMRCA schemes can be extended to support explicit rate mechanism.



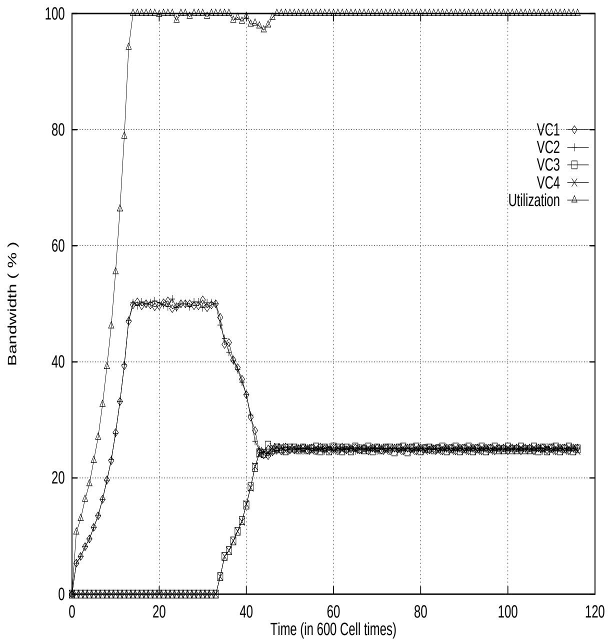
(a) EPRCA



(b) MMRCA_Basic

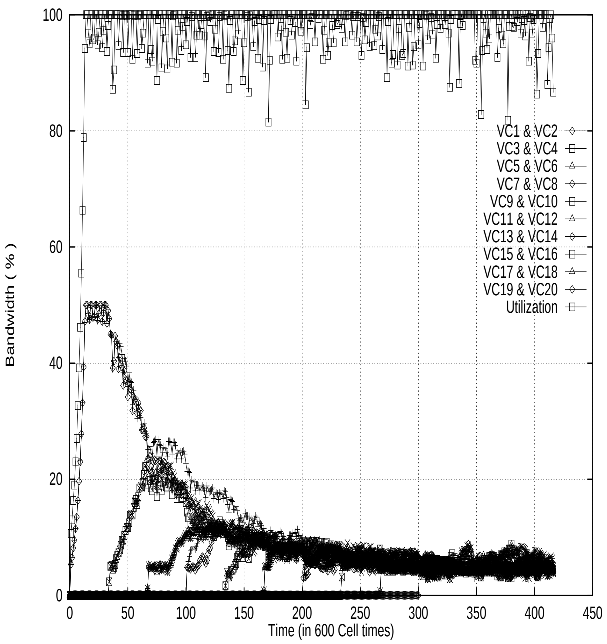


(c) MMRCA_RQL

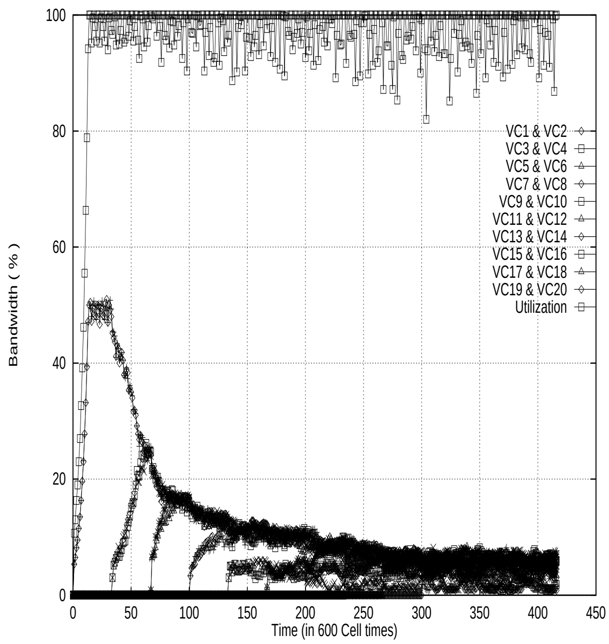


(d) MMRCA_TR

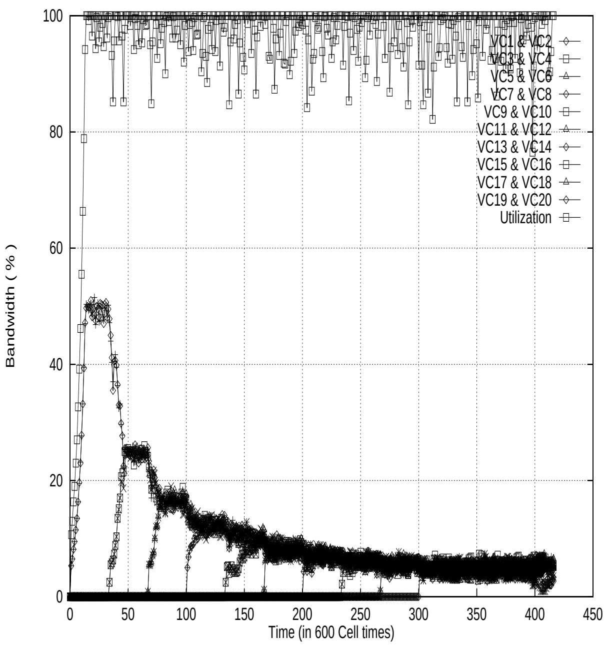
Figure 8: Instantaneous bandwidth of each VC and the total link utilization using EPRCA and MMRCA schemes for homogeneous non-bottlenecked LAN model with four links ($N = 4$). The total simulation time is 70000 cell time units. The average and maximum buffer size required are: (a) for EPRCA: 33 and 150, (b) for MMRCA_Basic: 23 and 83, (c) for MMRCA_RQL: 19 and 66, (d) for MMRCA_TR: 10 and 41.



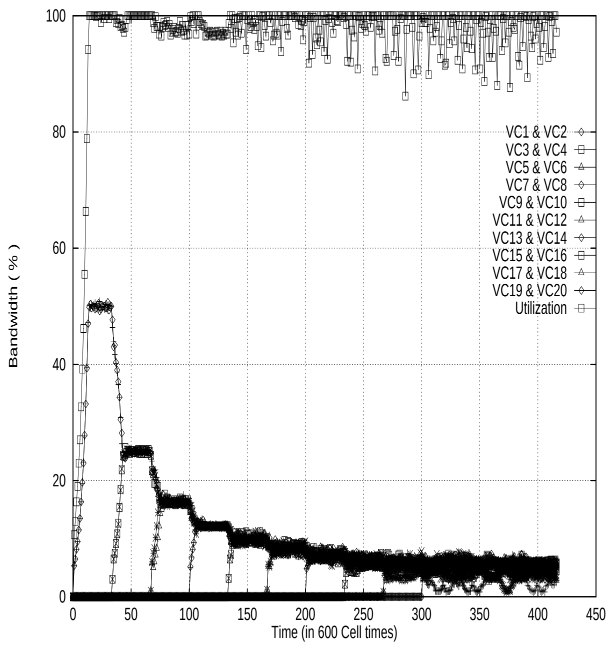
(a) EPRCA



(b) MMRCA_Basic



(c) MMRCA_RQL



(d) MMRCA_TR

Figure 9: Instantaneous bandwidth of each VC and the total link utilization using EPRCA and MMRCA schemes for homogeneous non-bottlenecked LAN model with 20 links ($N = 20$). The total simulation time is 250000 cell time units.

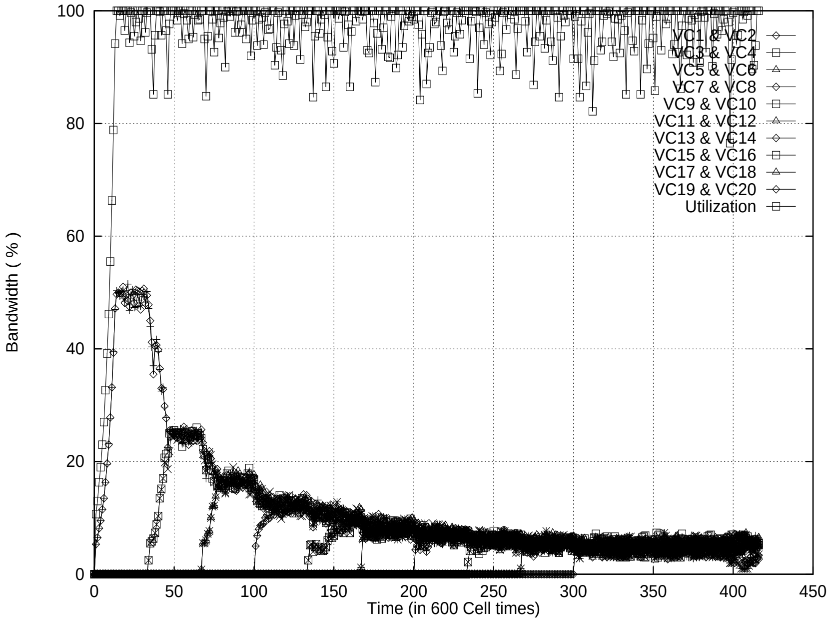
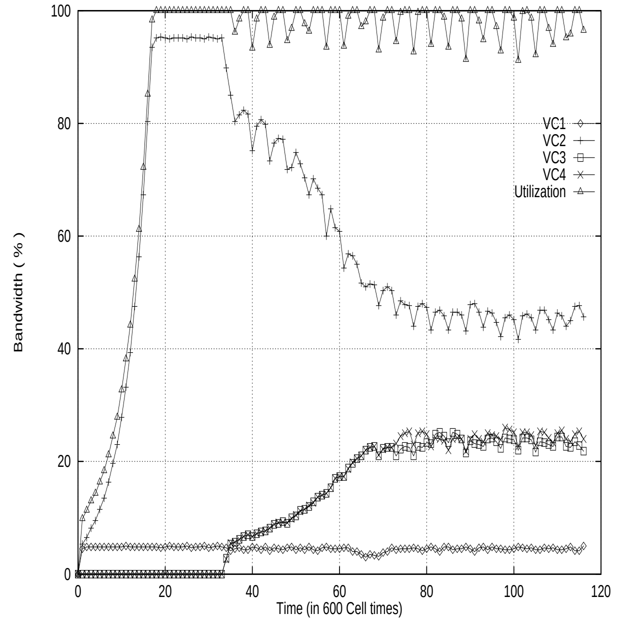
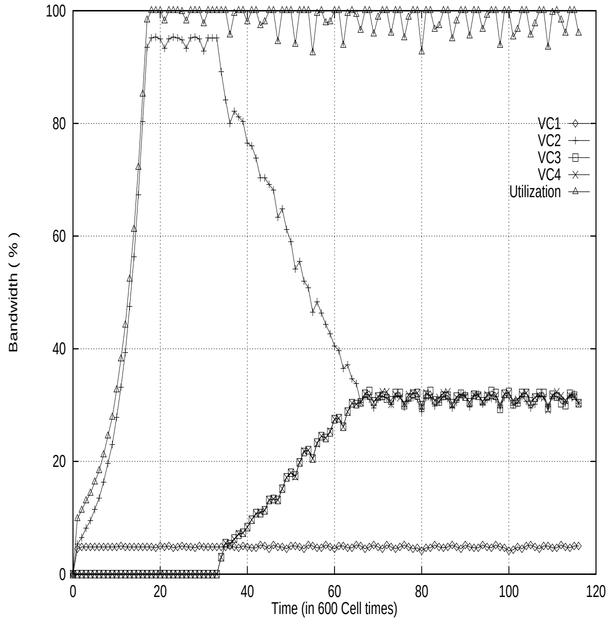


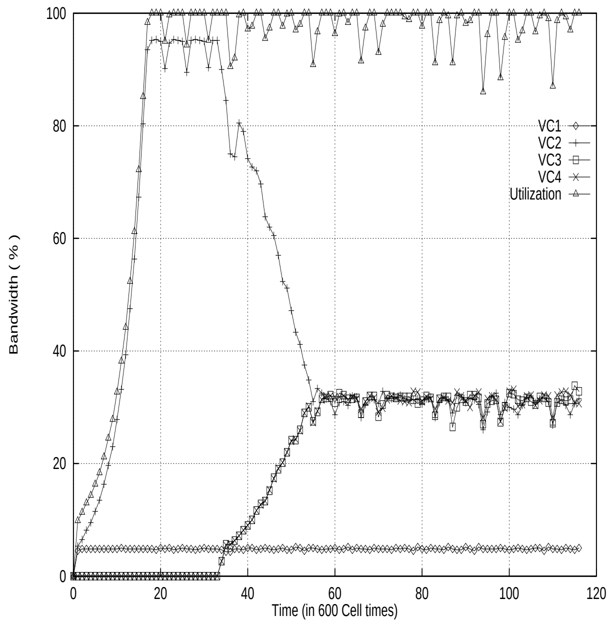
Figure 10: Instantaneous bandwidth of each VC and the total link utilization for MMRCA_RQL scheme with parameter $RQL = 0$. The average and maximum buffer size for this case are 13 and 107, and the link utilization is 93.2%.



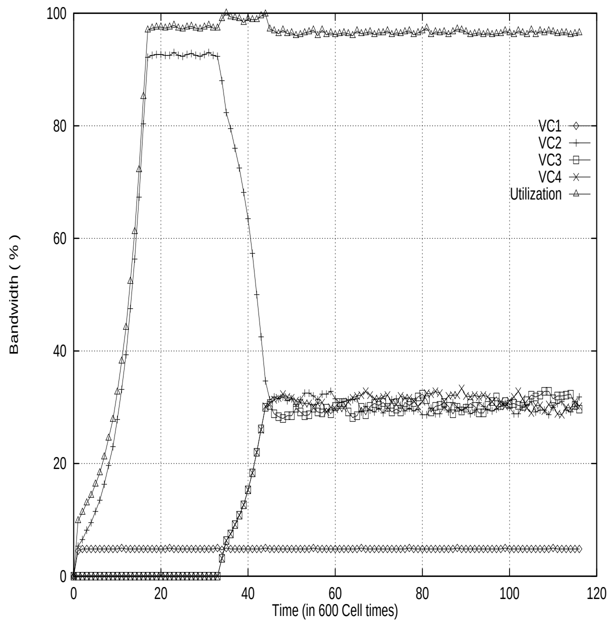
(a) EPRCA



(b) MMRCA_Basic



(c) MMRCA_RQL



(d) MMRCA_TR

Figure 11: Instantaneous bandwidth of each VC and the total link utilization using EPRCA and MMRCA schemes for homogeneous **bottlenecked** LAN model with four links ($N = 4$). The total simulation time is 70000 cell time units. The average and maximum buffer size required are: (a) for EPRCA: 30 and 103, (b) for MMRCA_Basic: 18 and 65, (c) for MMRCA_RQL: 15 and 71, (d) for MMRCA_TR: 1 and 22.

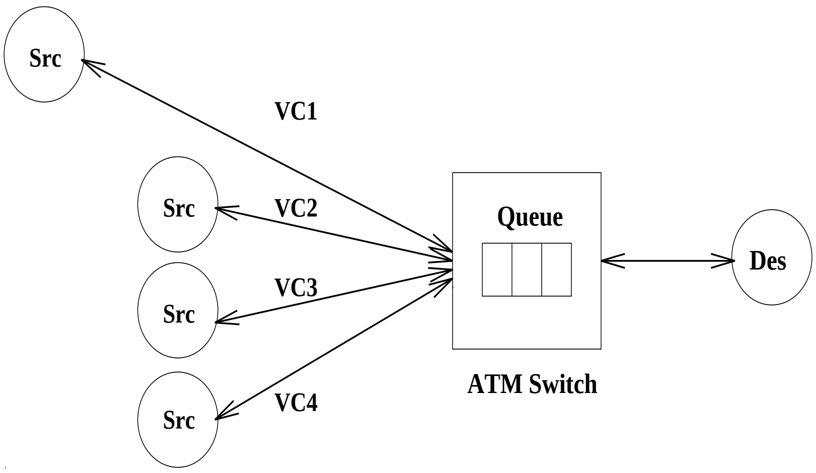
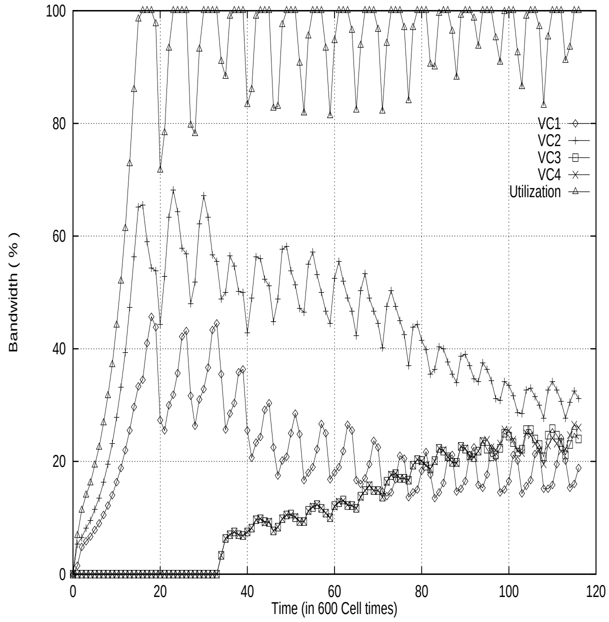
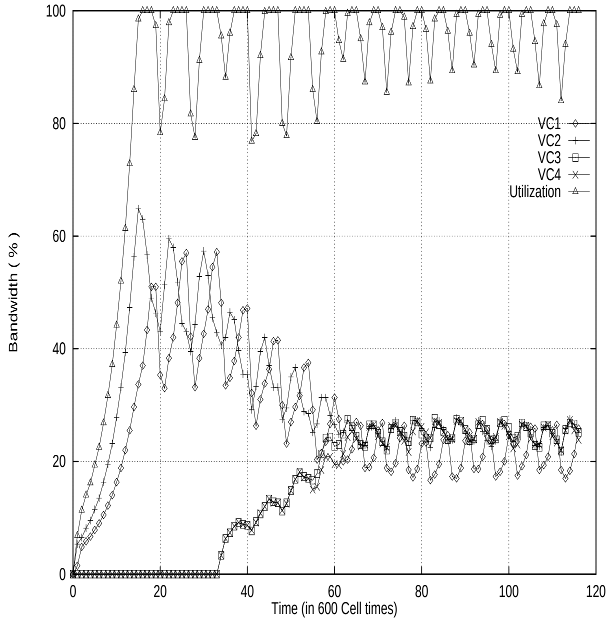


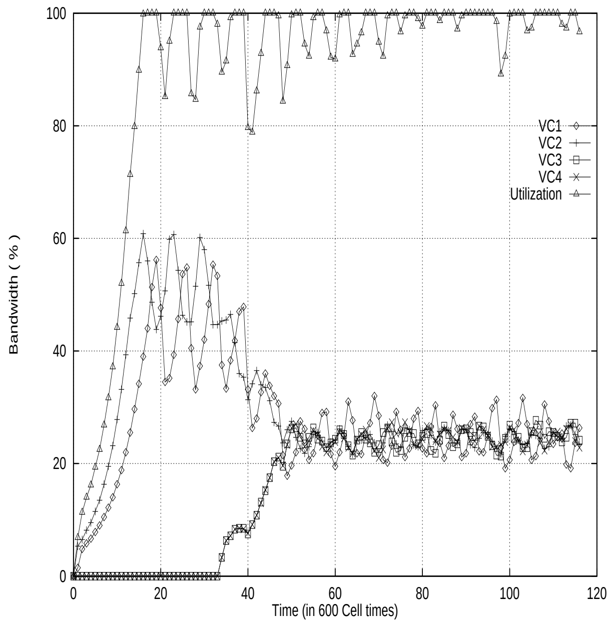
Figure 12: A simple wide area network (WAN) topology with one switch and four VCs. The VC1 link is 220 miles long and has a propagation delay of 400 cell time units. All other links are of the same length (22 miles) and have a propagation delay of 40 cell time units.



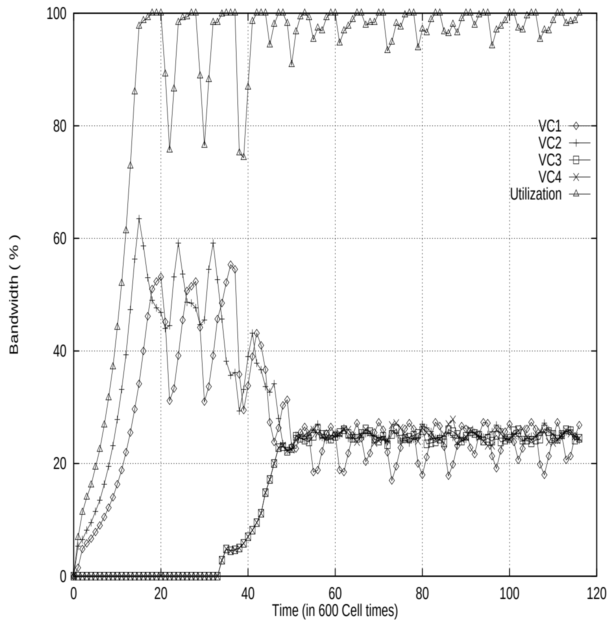
(a) EPRCA



(b) MMRCA_Basic



(c) MMRCA_RQL



(d) MMRCA_TR

Figure 13: Instantaneous bandwidth of each VC and the total link utilization using EPRCA and MMRCA schemes for homogeneous non-bottlenecked WAN model with four links ($N = 4$). The total simulation time is 70000 cell time units. The average and maximum buffer size required are: (a) for EPRCA: 43 and 153, (b) for MMRCA_Basic: 31 and 144, (c) for MMRCA_RQL: 18 and 94, (d) for MMRCA_TR: 13 and 130.

References

[1] I. S. Gradshteyn, and I. M. Ryzhik, "Table, Integrals, Series and Products", *Academic*, 1980, pp: xxviii, 2.

[2] B. A. Makrucki, "Explicit Forward Congestion Notification in ATM Networks", *Proc. TriComm*, February 1992.

[3] P. Newman, "Backward Explicit Congestion Notification for ATM Local Area Networks", *IEEE GLOBECOM*, December 1993, pp. 719-723.

[4] N. Yin, and M. Hluchyi, "On Closed-Loop Rate Controls for ATM Cell Relay Networks", *IEEE INFOCOM*, June 1994.

[5] L. Roberts et al., "Closed-loop Rate-Based Traffic Management", *ATM Forum Contribution 94-0438R1*, June 1994.

[6] J. C. R. Bennett and G. T. D. Jardins, "Comments on the July PRCA Rate Control Baseline", *ATM Forum Contribution 94-0682*, July 1994.

[7] J. C. R. Bennett and G. T. D. Jardins, "Failure Modes of the Baseline Rate Based Congestion Control Plans", *ATM Forum Contribution 94-0682*, July 1994.

[8] L. Roberts, "Enhanced PRCA (proportional rate control algorithm)", *ATM Forum Contribution 94-0735R1*, August 1994.

[9] H. Hsiaw et al., "Closed-loop Rate-based Traffic Management", *ATM Forum Contribution 94-0438R2*, September 1994.

[10] R. Jain, S. Kalyanaraman, and R. Viswanathan, "The OSU Scheme for Congestion Avoidance using Explicit Rate Indication", *ATM Forum Contribution 94-0883*, September 1994.

[11] K.-Y. Siu, and H.-Y. Tzeng, "Adaptive Proportional rate Control (APRC) with Intelligent Congestion Indication", *ATM Forum Contribution 94-0888*, September 1994.

[12] H. T. Kung and A. Chapman, " Credit-based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing", *Proc. SIGCOMM'94*, vol. 24, Oct. 1994, pp. 101-114.

[13] K.-Y. Siu, and H.-Y. Tzeng, "Intelligent Congestion Control for ABR Service in ATM Networks", *ACM SIGCOMM, Computer Communication review*, 1995.

[14] H. Ohsaki et al., "Rate-Based Congestion Control for ATM Networks", *ACM SIGCOMM, Computer Communication review*, Vol. 25 (2), April 1995.

[15] A. Charny, D. D. Clark, and R. Jain, "Congestion Control With Explicit Rate Indication", *Proc. ICC*, June 1995, pp. 1954-1963.

[16] S. Sathaye, "Traffic Management Specification Version 4.0", *ATM Forum Contribution 95-0013R7*, July 1995.

[17] R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey", *Computer Networks and ISDN Systems*, 1995.

[18] S. Muddu, F. Chiussi, and V. Kumar, "Generalized One-bit Congestion Control Algorithm", *manuscript*, Aug. 1995.

[19] A. Arulambalam., "Impact of Queuing Disciplines on Available Bit rate Congestion Control in ATM Networks", *Submitted to INFOCOM 1997*, July 1995.

Appendix A: Control Parameters

This appendix list control parameters and variables used by congestion control algorithms.

Source End System Parameters

PCR	Peak Cell Rate
MCR	Minimum Cell Rate
ICR	Initial Cell Rate
CCR	Current Cell Rate
ACR	Allowed Cell Rate; same as Current Cell Rate for a VC
AIR	Additive Increase Rate
ADR	Additive Decrease Rate; decrease indicated via RM cell
PDR	Periodic Decrease Rate; decrease in source rate to avoid network collapse, if RM cells are lost
N_{RM}	Number of Data Cell per RM Cell
N_P	Number of Data Cells Transmitted for Periodic Source Rate Decrease; after N_P cells ACR decreases by PDR
TOF	Time Out Factor; used in the source end system to control the periodic source rate decrease

Switch Parameters

$MACR$	Mean Allowed Cell Rates
MAX	Current Maximum Cell rate
MIN	Current Minimum Cell rate
IPF	Increase Pressure Factor
DPF	Decrease Pressure Factor

DQT	High Queue Threshold; to determine high congestion
QT	Queue Threshold; to determine congestion
N_{QL}	Number of Data Cells; used to determine rate of change of queue length
RQL	Threshold on Queue Length Change; maximum change in queue length in N_{QL} time units
MX_MN_DIFF	Threshold on Difference between Maximum and Minimum Cell Rates
MAX_VC	VC with Current Maximum rate
MIN_VC	VC with Current Minimum rate

RM Cell Fields

CCR	Current Cell Rate
DIR	Direction of RM Cell; forward or backward
CI	Congestion Indicator; $CI = 1$ congestion and $CI = 0$ no congestion
NI	No Increase indicator; $NI = 1$ no additive increase and $NI = 0$ additive increase
ER	Explicit Rate field; used to limit the source rate ACR to ER