RATE BASED CONGESTION CONTROL FOR ATM NETWORKS

INIVERSITY

Dissertation submitted in partial fulfilment of requirements for the award of the degree of

Master of Technology
in
Computer Science

by
Sanjeev Tiwari

ज.ने.वि.
JNU

SCHOOL OF COMPUTER & SYSTEM SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI - 110067

January 1998

CERTIFICATE

This is to certify that the dissertation entitled **RATE BASED CONGESTION CONTROL FOR ATM NETWORKS** being submitted by **Mr. Sanjeev Tiwari** to the **School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfilment of the requirements for the award of the degree of **Master of Technology** in **Computer Science**, is a bonafide work carried by him under the guidance and supervision of **Prof. C.P. Katti**.

The matter embodied in the dissertation has not been submitted for the award of any other degree or diploma.

Prof. C.P. Katti
Professor, SC&SS
Jawaharlal Nehru University
New Delhi - 110067

Sanjeev Tiwari
Sanjeev Tiwari

19/3
Prof. P.C. Saxena
Dean, SC&SS
Jawaharlal Nehru University
New Delhi - 110067

ACKNOWLEDGEMENT

I would like to express my profound gratitude and personal regards to my guide **Prof. C.P. Katti** who provided the primary incentives for development of this project. I am very thankful for his time to time questions, suggestions and constructive criticism which shaped the project design and development.

I am indebted to **Prof. P.C. Saxena**, Dean, **School of Computer and System Sciences**, for providing me with the inspiration and necessary facilities in the centre for successful completion of the project.

It is my pleasure to acknowledge the assistance of many friends and colleagues who encouraged me while I was doing my project which contributed measurably to the quality of my thinking.

Last, but not least, my friend Soheli gave me lot of emotional support without which I could never have succeeded. Whenever I needed some encouragement, she some how sensed that and provided it at just the right time.

Sanjeev Tiwari

CONTENTS

LIST OF FIGURES

ABSTRACT

Congestion control is important in high speed networks. Due to larger bandwidth distance product, the amount of data lost due to simultaneous arrivals of burst from multiple sources can be larger. For the success of ATM, it is important that it provides a good traffic management for both bursty and non-bursty sources.

Here, concepts in congestion control for ATM networks are explained. The specification for ATM traffic control proposed by ATM forum are presented and evolution of rate-based framework for ABR (Available Bit Rate) servicve have been discussed. Some rate-based congestion control schemes have been described and compared. In the end, analysis of ERICA (Explicit Rate Indication for Congestion Avoidance), a rate-based congestion control scheme has been given along with some possible modification. Pseudocode for the above algorithm is also given.

CHAPTER 1

INTRODUCTION

The future telecommunication should have such characteristics: broadband, multimedia, economical implementation for diversity of services. Broadband integrated services digital networks (B-ISDN) provides what we need. Asynchronous Transfer Mode (ATM) is a target technology for meeting these requirements. In ATM networks, the information is transmitted using shortfixed-length cells, which reduces the delay variance, making it suitable for integrated traffic consisting of voice, video and data. By proper traffic management, ATM can also ensure efficient operation to meet different quality of service (QoS) desired by different types of traffic.

1.1 How ATM Works

1. ATM network uses fixed-length cells to transmit information. The cell consists of 48 bytes of payload and 5 bytes of header. The flexibility needed to support variable transmission rates is provided by transmitting the necessary number of cells per unit time.
2. ATM network is connection-oriented. It sets up virtual channal connection (VCC) going through one or more virtual paths (VP) and virtual channals (VC) before transmitting information. The cells is switched according to the VP or VC identifier (VPI/VCI) value in the cell head, which is originally set at the connection setup and is translated into new VPI/VCI value while the cell passes each switch.
3. ATM resources such as bandwidth and buffers are shared among users, they are allocated to the user only when they have something to transmit. So the network uses statistical multiplexing to improve the effective thoughput.

1.2 Need for congestion control

The assumption that statistical multiplexing can be used to improve the link utilization is that the users do not take their peak rate values simultaneously. But since the traffic demands are stochastic and cannot be predicted, congestion is unavoidable. Whenever the total input rate is greater than the output link capacity, congestion happens. Under a congestion situation, the queue length may become very large in a short time, resulting in buffer overflow and cell loss. So congestion control is necessary to ensure that users get the negociated QoS.

1.3 Misconception regarding congestion control

There are several misunderstandings about the cause and the solutions of congestion control.

1. Congestion is caused by the shortage of buffer space. The problem will be solved when the cost of memory becomes cheap enough to allow very large memory.

   Larger buffer is useful only for very short term congestions and will cause undesirable long delays. Suppose the total input rate of a switch is 1Mbps and the capacity of the output link is 0.5Mbps, the buffer will overflow after 16 second with 1Mbyte memory and will also overflow after 1 hour with 225Mbyte memory if the situation persists. Thus larger buffer size can only postpone the discarding of cells but cannot prevent it. The long queue and long delay introduced by large memory is undesirable for some applications.

2. Congestion is caused by slow links. The problem will be solved when high-speed links become available.

   It is not always the case, sometimes increases in link bandwidth can aggravate the congestion problem because higher speed links may make the network more unbalanced. For the configuration, if both of the two sources begin to send to destination at their peak rate, congestion will occur at the switch. Higher speed links can make the congestion condition in the switch worse.

3. Congestion is caused by slow processors. The problem will be solved when processor speed is improved.

This statement can be explained to be wrong similarly to the second one. Faster processors will transmit more data in unit time. If several nodes begin to transmit to one destination simultaneously at their peak rate, the target will be overwhelmed soon. Congestion is a dynamic problem, any static solutions are not sufficient to solve the problem. All the issues presented above: buffer shortage, slow link, slow processor are symptoms not the causes of congestion. Proper congestion management mechanisms is more important than ever.

1.4 Expectation from Congestion Control

1.4.1 Objectives

The objectives of traffic control and congestion control for ATM are: Support a set of QoS parameters and classes for all ATM services and minimize network and end-system complexity while maximizing network utilization.

1.4.2 Selection Criteria

To design a congestion control scheme is appropriate for ATM network and non-ATM networks as well, the following guidances are of general interest.

* *Scalability*

The scheme should not be limited to a particular range of speed, distance, number of switches, or number of VCs. The scheme should be applicable for both local area networks (LAN) and wide area networks (WAN).

* *Fairness*

In a shared environment, the throughput for a source depends upon the demands by other sources. There are several proposed criterion for what is the correct share of bandwidth for a source in a network environment. And there are ways to evaluate a bandwidth allocation scheme by comparing its results with a optimal result.

* *Fairness Criteria*

1. Max-Min

   The available bandwidth is equally shared among connections.

2. MCR plus equal share

   The bandwidth allocation for a connection is its MCR plus equal share of the available bandwidth with used MCR removed.

   The nth active connection's rate B_n is given by

$$B_n = MCR_n + \frac{C - \Sigma_{i=1}^{N_{vc}} MCR_i}{N_{VC}}$$

$$1 \leq n \leq N_{VC}$$

3. Maximum of MCR or Max-Min share

   The bandwidth allocation for a connection is its MCR or Max-Min share, which ever is larger. In this definition, each connection acquires larger bandwidth of MCR and the bandwidth equally divided by all connections :

$$B_n = \max(\frac{C}{N_{VC}}, MCR_n)$$

$$1 \leq n \leq N_{VC}$$

This assignment needs an iteration for the sum of Bn's to be settled at the available bandwidth C, and the required number of iterations cannot be estimated. For connections with larger MCR, however, more bandwidth can be allocated than in the previous case.

4. Allocation proportional to MCR

The bandwidth allocation for a connection is weighted proportional to its MCR The definition assigns the avialable bandwidth to unconstrained connections in weighted manner as :

$$B_n = C \frac{MCR_n}{\Sigma_{i=1}^{N_{vc}} \; MCR_i}$$

$$1 \leq n \leq N_{VC}$$

5. Weighted allocation

The bandwidth allocation for a connection is proportional to its pre-determined weight. This is a hybrid of the second and fourth :

$$B_n = MCR_n + F_n(C - \Sigma_{i=1}^{N_{vc}} \; MCR_i),$$

$$1 \leq n \leq N_{VC}$$

Where F_n is a weight for the nth connection and is defined as

$$F_n = \frac{b}{N_{VC}} + (1 - b) \frac{MCR_n}{\Sigma_{i=1}^{N_{vc}} \; MCR_i}$$

$$1 \leq n \leq N_{VC}$$

* ***Fairness Index***

The share of bandwidth for each source should be equal to or converge to the optimal value according to some optimality criterion. We can estimate the fairness of a certain scheme numerically as follows. Suppose a scheme allocates x1, x2,..., xn, while the optimal allocation is y1, y2, ..., yn. The normalized allocation is zi = xi / yi for each source and the fairness index is defined as following:

Fairness = sum(zi) * sum(zi) / sum(zi * zi)

- *Robustness*

The scheme should be insensitive to minor deviations such as slight mistuning of parameters or loss of control messages. It should also isolate misbehaving users and protect other users from them.

- *Implementability*

The scheme should not dictate a particular switch architecture. It also should not be too complex both in term of time or space it uses.

1.5 Connection Parameters

1.5.1 Quality of Service

A set of parameters are negotiated when a connection is set up on ATM networks. These parameters are used to measure the Quality of Service (QoS) of a connection and quantify end-to-end network performance at ATM layer. The network should guarantee the QoS by meet certain values of these parameters.

- *Cell Transfer Delay (CTD):*

The delay experienced by a cell between the first bit of the cell is transmitted by the source and the last bit of the cell is received by the destination. Maximum Cell Transfer Delay (Max CTD) and Mean Cell Transfer Delay (Mean CTD) are used.

*   *Peak-to-peak Cell Delay Variation (CDV):*

The difference of the maximum and minimum CTD experienced during the connection. Peak-to-peak CDV and Instantaneous CDV are used.

*   *Cell Loss Ratio (CLR):*

The percentage of cells that are lost in the network due to error or congestion and are not received by the destination.

1.5.2 Usage Parameters

Another set of parameters are also negotiated when a connection is set up. These parameters discipline the behavior of the user. The network only provide the QoS for the cells that do not violate these specifications.

*   ***Peak Cell Rate (PCR):***

The maximum instantaneous rate at which the user will transmit.

*   ***Sustained Cell Rate (SCR):***

The average rate as measured over a long interval.

*   ***Burst Tolerance (BT):***

The maximum burst size that can be sent at the peak rate.

*   ***Maximum Burst Size (MBS):***

The maximum number of back-to-back cells that can be sent at the peak cell rate. BT and MBS are related as follows: Burst Tolerance = (MBS - 1)(1/SCR - 1/PCR)

*　*Minimum Cell Rate (MCR):*

The minimum cell rate desired by a user.

1.6 Service Categories

Providing desired QoS for different applications is very complex. For example, voice is delay-sensitive but not loss-sensitive, data is loss- sensitive but not delay-sensitive, while some other applications may be both delay-sensitive and loss-sensitive.

To make it easier to manage, the traffic in ATM is divided into five service classes:

*　***CBR: Constant Bit Rate***

Quality requirements: constant cell rate, i.e. CTD and CDV are tightly constrained; low CLR.

Example applications: interactive video and audio.

*　***rt-VBR: Real-Time Variable Bit Rate***

Quality requirements: variable cell rate, with CTD and CDV are tightly constrained; a small nonzero random cell loss is possible as the result of using statistical multiplexing.

Example applications: interactive compressed video.

*　***nrt-VBR: Non-Real-Time Variable Bit Rate***

Quality requirements: variable cell rate, with only CTD are tightly constrained; a small nonzero random cell loss is possible as the result of using statistical multiplexing.

Example applications: response time critical transaction processing.

*　***UBR: Unspecified Bit Rate***

Quality requirements: using any left-over capacity, no CTD or CDV or CLR constrained.

Example applications: email and news feed.

* *ABR: Available Bit Rate*

Quality requirements: using the capacity of the network when available and controlling the source rate by feedback to minimize CTD , CDV and CLR.

Example applications: critical data transfer, remote procedure call and distributed file service.

These service categories relate traffic characteristics and QoS requirements to network behaviour. The QoS requirement for each class is different. The traffic management policy for them are different, too.

Among these service classes, ABR is commonly used for data transmissions which require a guaranteed QoS, such as low probability of loss and error. Small delay is also required for some application, is not as strict as the requirement of loss and error. Due to the burstiness, unpredictability and huge amount of the data traffic, congestion control of this class is the most needed and is also the most studied.

ATM Forum Technical Committee specified the feedback mechanism for ABR flow control. We will discuss it in more detail later.

CHAPTER 2

GENERIC FUNCTIONS

It is observed that events responsible for congestion in broadband networks have time constants that differ by orders of magnitude, and multiple controls with appreciate time constants are necessary to manage network congestion.

We can **classify the congestion control schemes by the time scale** they operate upon:

1) network design
2) connection admission control (CAC)
3) routing (static or dynamic)
4) traffic shaping
5) end-to-end feedback control
6) hop-by-hop feedback control
7) buffering

The different schemes are functions on different severity of congestion as well as different duration of congestion.

Another **classification of congestion control schemes is by the stage** that the operation is performed:

1) congestion prevention
2) congestion avoidance
3) congestion recovery

Congestion prevention is the method that make congestion impossible. Congestion avoidance is that the congestion may happen, but the method avoid it by get the network state always in balance. Congestion recovery is the remedy steps to take to pull the system out of the congestion state as soon as possible and make it less damaging when the congestion already happened.

No matter what kind of scheme is used, the following outstanding problems are the main difficulties that need to be treated carefully:

1) The burstiness of the data traffic.

2) The unpredictability of the resource demand.

3) The large propagation delay verses the large bandwidth.

To meet the objectives of traffic control and congestion control in ATM networks, the following functions and procedures are suggested by the ATM Forum Technical Committee.

2.1 Connection Admission Control

Connection Admission Control (CAC) is defined as the set of actions taken by the network during the call set-up phase in order to determine whether a connection request can be accepted or should be rejected.

2.2 Usage Parameter Control

Usage Parameter Control (UPC) is defined as the set of actions taken by the network to monitor and control traffic at the end-system access. Its main purpose is to protect network resources from user misbehavior, which can affect the QoS of other connections, by detecting violations of negotiated parameters and taking appropriate actions.

2.3 Generic Cell Rate Algorithm

The Generic Cell Rate Algorithm (GCRA) is used to define conformance with respect to the traffic contract. For each cell arrival, the GCRA determines whether the cells conforms to traffic contract of the connection. The UPC function may implement GCRA, or one or more equivalent algorithms to enforce conformance.

GCRA is a virtual scheduling algorithm or a continuous-state Leaky Bucket Algorithm. The GCRA is used to define the relationship between PCR and CDVT, and relationship between SCR and BT. The GCRA is also used to specify the conformance of the declared values of and the above parameters.

2.4 Priority Control

The end-system may generate traffic flows of different priority using the Cell Loss Priority (CLP) bit. The network may selectively discard cells with low priority if necessary such as in congestion to protect, as far as possible, the network performance for cells with high priority.

2.5 Traffic Shaping

Traffic shaping is a mechanism that alters the traffic characteristics of a stream of cells on a connection to achieve better network efficiency whilst meeting the QoS objectives, or to ensure conformance at a subsequent interface.

Examples of traffic shaping are peak cell rate reduction, burst length limiting, reduction of CDV by suitably spacing cells in time, and queue service schemes. Traffic shaping may be performed in conjunction with suitable UPC functions.

2.6 Network Resource Management

In Network Resource Management (NRM) is responsible for the allocation of network resources in order to separate traffic flows according to different service characteristics, to maintain network performance and to optimise resource utilisation. This function is mainly concerned with the management of virtual paths in order to meet QoS requirements.

2.7 Frame Discard

If a congested network needs to discard cells, it may be better to drop all cells of one frame than to randomly drop cells belonging to different frames, because one cell loss may cause the retransmission of the whole frame, which may cause more traffic when congestion already happened. Thus, frame discard may help avoid congestion collapse and can increase throughput. If done selectively, frame discard may also improve fairness.

2.8 Feedback Control

Feedback controls are defined as the set of actions taken by the network and by the end-systems to regulate the traffic submitted on ATM connections according to the state of network elements.

Feedback mechanisms are specified for ABR service class by ATM Forum Technical Committee. We will discuss it in detail later.

2.9 ABR Flow Control

As we have discussed before, the ABR service category uses the link capacity that is left over and is applied to transmit critical data that is sensitive to cell loss. That makes traffic management for this class the most challenging by the fluation of the network load condition, the burstiness of the data traffic itself, and the CLR requirement.

The ATM Forum Technical Committee Traffic Management Working Group have worked hard on this topic, and here are some of the main issues and the current progress of this area.

Congestion management in ATM is a hotly debated topic, many contradictory beliefs exist on most issues. These beliefs lead to different approaches in the congestion control schemes. Some of the approaches are :

1. Open-Loop vs. Close-Loop

Open-loop approaches do not need end-to-end feedback, one of the examples of this type are prior-reservation and hop-to-hop flow control. In close-loop approaches, the source adjust its cell rate in responding to the feedback information received from the network.

It has been argued that close-loop congestion control schemes are too slow in todays high-speed, large range network, by the time a source gets the feed back and reacts to it, several thousand cells may have been lost. But on the other hand, if the congestion has already happened and and the overload is of long duration, the condition cannot be released unless the source causing the congestion is asked to reduce its rate. Furthermore, ABR service is designed to use any bandwidth that is left over the source must have some knowledge of what is available when it is sending cells. The ATM Forum Technical Committee Traffic Management Working Group specified that feedback is necessary fro ABR flow control.

2. *Credit-Based vs. Rate-Based*

Credit-Based approaches consists of per-link, per-VC window flow control. The receiver monitors queue lengths of each VC and determines the number of cells the sender can transmit on that VC, which is called "credit". The sender transmits only as many cells as allowed by the credit.

Rate-Based approaches control the rate by which the source can transmit. If the network is light loaded, the source are allowed to increase its cell rate. If the network is congested, the source should decrease its rate.

After a long debate, ATM Forum finally adopted the rate-based approach and rejected the credit-based approach. The main reason for the credit-based approach not being adopted is that it requires per-VC queuing, which will cause considerable complexity in the large switches which support millions of VCs. It is not scalable. Rate-Based approaches can work with or without per-VC queuing.

3. *Binary Feedback vs. Explicit Feedback*

Binary Feedback uses on bit in the cell to indicate the elements along the flow path is congested or not. The source will increase or decrease its rate by some pre-decided rule upon receive the feedback. In Explicit Feedback, the network tells the source exactly what rate is allowed for it to send.

Explicit Rate (ER) feedback approach is preferred, because ER schemes have several advantages over single-bit binary feedback First, ATM networks are connection oriented and the switches know more information along the flow path, the increased information can only be used by explicit rate feedback. Secondly, the explicit rate feedback is faster to get the source to the optimal operating point. Third, policing is straight forward. The entry switches can monitor the returning message and use the rate directly. Fourth, with fast convergence time, the initial rate has less impact. Fifth, the schemes are robust against errors in or loss of a single message. the next correct message will bring the system to the correct operating point. There are two ways for explicit rate feedback: forward feedback and backward feedback.

With forward feedback, the messages are sent forward along the path and are returned to the source by the destination upon receiving the message. With backward feedback, the messages are sent directly back to the source by the switches whenever congestion condition happens or is pending in any of the switches along the flow path.

4. Congestion Detection

Queue Length vs. Queue Growth Rate Actually this issue does not cause too much debate. In earlier schemes, large queue length is often used as the indication of congestion. But there some problems with this method. First, it is a static measurement. For example, a switch with a 10k cells waiting in queue is not necessarily more congested than a switch with a 10 cell queue if the former one is draining out its queue with 10k cell per second rate and the queue in the latter is building up quickly. Secondly, using queue length as the method of congestion detection was shown to result in unfairness. Sources that start up late were found to get lower throughput than those which start early. Queue growth rate is more appropriate as the parameter to monitor the congestion state because it shows the direction that the network state is going. It is natural and direct to use queue growth rate in a rate-based scheme, with the controlled parameter and the input parameter have the same unit.

CHAPTER 3

RATE-BASED CONGESTION CONTROL FRAMEWORK

3.1 RM-cell Structure

In the ABR service, the source adapts its rate to changing network conditions. Information about the state of the network like bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special control cells called Resource Management Cells (RM-cells).

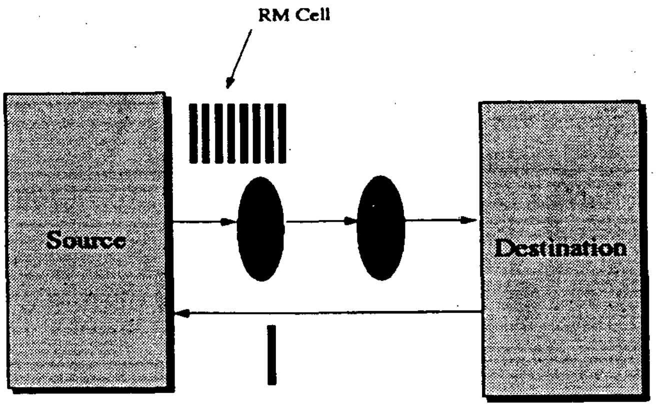

Figure 1: RM cell path

ATM Forum Technical Committee specifies the format of the RM-cell. The already defined fields in a RM-cell that is used in ABR service is explained in this section.

1. Header

   The first five bytes of an RM-cell are the standard ATM header with PTI=110 for a VCC and VCI=6 for a VPC.

2. ID

   The protocol ID. The ITU has assigned this field to be set to 1 for ABR service.

3. DIR

Direction of the RM-cell with respect to the data flow which it is associated with. It is set to 0 for forward RM-cells and 1 for backward RM-cells.

4. BN

Backward Notification. It is set to 1 for switch generated (BECN) RM-cells and 0 for source generated RM-cells.

5. CI

Congestion Indication. It is set to 1 to indicate congestion and 0 otherwise.

6. NI

No Increase. It is set to 1 to indicate no additive increase of rate allowed when a switch senses impending congestion and 0 otherwise.

7. ER

Explicit rate. It is used to limit the source rate to a specific value.

8. CCR

Current Cell Rate. It is used to indicate to current cell rate of the source.

9. MCR

Minimum Cell Rate. The minimum cell rate desired by the source.

3.2 Service Parameters

ATM Forum Technical Committee defined a set of flow control parameters for ABR service.

1. PCR

Peak Cell Rate, it is the source desired but the maximum rate the network can support. It is negotiated when the connection is set up.

2. MCR

Minimum Cell Rate, the source need not reduce its rate below it under any condition.

It is negotiated when the connection is set up.

3. ICR

Initial Cell Rate, the startup rate after idle periods. It is negotiated when the connection is set up.

4. AIR

Additive Increase Rate, the highest rate increase possible. It is negotiated when the connection is set up.

5. Nrm

The number of cells transmitted per RM-cell sent. It is negotiated when the connection is set up.

6. Mrm

Used by the destination to control allocation of bandwidth between forward RM-cells, backward RM-cells, and data cells. It is negotiated when the connection is set up.

7. RDF

Rate Decrease Factor, to control the number of cells sent upon idle startup before the network can establish control in one Round Trip Time (RTT). It is negotiated when the connection is set up.

8. ACR

Allowed Cell Rate, the source can not transmit with rate higher than it.

9. Xrm

The maximum RM-cells sent without feedback before the source need to reduce its rate. It is negotiated when the connection is set up.

10. TOF

Time Out Factor, to control the maximum time permitted between sending forward RM-cells before a rate decrease is required. It is negotiated when the connection is set up.

11. Trm

The inter-RM time interval used in the source behavior. It is negotiated when the connection is set up.

12. RTT

Round Trip Time between the source and the destination. It is computed during call setup.

13. XDF

Xrm Decrease Factor, specify how much ofthe reduction ofthe source rate when XRM is triggered. It is negotiated when the connection is set up.

These parameters are used to implement ABR flow-control on a per-connection basis, and the source, switch and destination must behave within the rules that defined by these parameters.

The function and usage of these parameters are still under study.

Source, Destination and Switch Behaviour

ATM Forum Technical Committee also specifies the source, destination, and switch behavior for the service.

There are two notations that need to be explained before we discuss the network behavior.

In-Rate Cells : The cells that counted in the user's rate with CLP=0. In-rate cells include data cells and in-rate RM-cells.

Out-of-Rate Cells : These cells are RM-cells and are not counted in the user's rate. They are used when ACR=0 and in-rate RM cells can not be send. The CLP is set to 1 for them.

In this section, we discuss some highlights of the specification.

3.3 Source Behaviour

1. The value of ACR shall never exceed PCR, nor shall it ever be less than MCR. The source shall never send in-rate cells at a rate exceeding ACR.

2. The source shall start with ACR at ICR and the first in-rate cell sent shall be a forward RM-cell.

3. The source shall send one RM-cell after every Nrm data cells.

4. If the source does not receive any feedback since it sends the last RM-cell, it shall reduce its rate by at least ACR*T*TDF after TOF*Nrm cell intervals.

5. If at least Xrm in-rate forward RM-cells have been sent since the last backward RM-cell with BN=0 was received, ACR shall be reduced by at least ACR*XDF.

6. When a backward RM-cell is received with CI=1, ACR shall be reduced by at least ACR*Nrm/RDF. If the backward RM-cell has both CI=0 and NI=0, the ACR may be increased by no more than AIR*Nrm.

7. Out-of-rate forward RM-cells shall not be sent at a rate greater than TCR.

3.4 Destination Behaviour

1. When a data cell is received, the destination shall save the EFCI state.

2. When returning an RM-cell, it shall set CI if saved EFCI is 1. Congested destination may set both CI and NI, or reduce ER.

3. If an RM-cell has not been returned while the next one arrives, throw away the old one.

4. The destination can generate a backward RM-cell without having received a forward RM-cell.

3.5 Switch Behaviour

1. The switch may set the EFCI flag in the data cell headers.

2. The switch may set CI or NI in the RM-cells, or may reduce the ER field.

3. The switch may generate backward RM-cells with CI or NI set.

CHAPTER 4

RATE BASED CONGESTION CONTROL REPRESENTATIVE SCHEMES

The following is a brief description of congestion control schemes that are proposed to the ATM Forum. The various mechanisms can be can be classified broadly depending upon the congestion monitoring criteria used and the feedback mechanism employed.

4.1 EFCI Control Schemes

These class of feedback mechanisms use binary feedback involving the setting of the EFCI bit in the cell header. The simplest example of a binary feedback mechanism is based on the old DECbit scheme. In this scheme all the VCs in a switch share a common FIFO queue and the queue length is monitored. When the queue length exceed a threshold congestion is declared and the cells passing the switch have their EFCI bit set. When the queue length falls below the threshold the cells are passed without their EFCI bit set. The source will adjust its rate accordingly when it sees the feedback cells with the EFCI bit set or not. Variations of this scheme include using two thresholds for the indication and removing congestion respectively. Binary feedback mechanisms can sometimes be fair because long hop VCs have higher possibility to have their cell EFCI bit set and get fewer opportunities to increase their rate. It is called the

"beat down problem". This problem can be alleviated by some enhancements to the basic scheme such provide separate queues for each VCs. But a coherent problem with binary feedback mechanisms are that they are too slow for rate-based control in high-speed networks.

4.2 Explicit Rate Feedback Schemes

As we have discussed before, explicit rate feedback control would not only be faster but would offer more flexibility to switch designers. Many explicit rate feedback control schemes has been proposed, the following are some that is documented by the ATM Forum.

4.2.1 Enhanced Proportional Rate Control Algorithm (EPRCA)

In EPRCA, the source sends data cells with EFCI set to 0 and sends RM-cells every n data cells. The RM-cells contain desired explicit rate (ER), current cell rate (CCR) and congestion indication (CI). The source usually initailizes CCR to the allowed cell rate (ACR) and CI to zero. The switch computes a mean allowed cell rate (MACR) for all VCs using exponential weighted average:

MACR = (1 - alpha) * MACR + alpha * CCR

and the fair share as a fraction of this average, where alpha and the fraction are chosen to be 1/16 and 7/8 respectively.

The ER field in the returning RM-cells are reduced to fair share if necessary. The switch may also set the CI bit in the cells passing when it is congested which is sensed by monitoring its queue length. The destination monitors the EFCI bits in data cells and mark the CI bit in the RM-cell if the last seen data cell had EFCI bit set.

The source decreases its rate continuously after every cell by a fixed factor and increases its rate by an fixed amount if the CI bit is not set. Another rule is that the new increased rate must never exceed either the ER in the returned cell or the PCR of the connection.

In EPRCA the fairness could be achieved if each connection is maintained separately at the switch, which is called per-VC accounting. However, since it requires an additional control complexity, EPRCA adopts another method "intelligent marking".The other is the means for reducing the rate of each connection explicitly; that is, the switch can have a responsibility for determining the cell transmission rate of selected connections. While some modifications were were required in order to The fairness could be achieved if each connection is maintained separately at the switch, which is called per-VC accounting. However, since it requires an additional control complexity, EPRCA adopts another method "intelligent marking". The other is the means for reducing the rate of each connection explicitly; that is, the switch can have a responsibility for determining the cell transmission rate of selected connections. While some modifications were were required in order to incorporate these new features, EPRCA preserves a backward compatibility with PRCA. A switch supporting only PRCA can thus also be used in an EPRCA-based network.

EPRCA requires forward RM cells as well as backward RM cells. RM cells contain a CI(Congestion Indication) bit that is used to carry congestion information to the source. Instead of unmarking an EFCI bit of data cells as PRCA does, the source end system periodically sends a forward RM cellevery N(rm) data cells. When the destination end system receives the forward RM cell,it returns the RM cell to the source as a backward RM cell. When doing this, the destination end system sets the CI bit of the backward RM cell according to the EFCI status of the last incoming data cell. The source end system can thus be notified of the congestion dected at the intermediate switches by marking the EFCI bit of the data cells in the forward path.

The two major enhancements of EPRCA - intelligent marking and explicit rate setting — require additional information fields in each RM cell: CCR(Current Cell Rate) and ER (Explicit Rate) fields. An ER element is used to decrease the source rate explicitly, and is initially set to PCR by the source

The main problem of this scheme is that the congestion detection is based on the queue length and this method is shown to result in unfairness. Sources that start up late may get lower throughput than those start early.

4.2.2 Target Utilization Band (TUB) Congestion Avoidance Scheme

In each switch, a target rate is defined as slightly below the link bandwidth, such as 85-90% of the full capacity. The input rate of the switch is measured over a fixed averaging interval. The load factor z is then computed as: Load Factor z = Input Rate / Target Rate When the load factor is far from z, which means the switch is either highly overloaded or highly underloaded, all VCs are asked to change their load by this factor z. When the load factor is close to 1, between 1-delta and 1+delta for a small delta, the switch gives different feedback to underloading sources and overloading sources. A fairshare is computed, and all sources whose rates are more than the fair share are asked to devided their rates by z/(1+delta), while those below the fair share are asked to devided their rates by z/(1-delta).

4.2.3 Explicit Rate Indication for Congestion Avoidance (ERICA)

This scheme tries to achieve efficiency and fairness concurrently by allowing underloaded VCs to increase their rate to fair share inspite of the conditions of the network and the sources already equal or greater than fair share may increase their rate if the link is under used. And the target capacity of the switch is set higher, 90-95% of the full bandwidth.

The switch calculates fair share as:

Fairshare = Target capacity / Number of active VCs

And the remaining capacity that a source can use is:

VCshare = CCR / Load Factor z

Then the switch sets the source's rate to the maximum of the two. The information used to compute the quantities comes from the forward RM-cells and the feedback is given in the backward RM-cells. this ensures that the most current infermation is used to provide fastest feedback. Another advantage of this scheme is that it has few parameters which can be tuned easily.

4.2.4 Congestion Avoidance Using Proportional Control (CAPC)

In this scheme, the switches also set a target utilization slightly below 1 and measure the input rate to compute load factor z. During underload (z is less than 1), fair share is increased as:

Fairshare = Fairshare * Min(ERU, 1+(1-z)*Rup)

where Rup is a slope parameter between 0.025 and 0.1, and ERU is the maximum increase allowed. During overload (z is greater than 1), fair share is decreased as:

Fairshare = Fairshare * Max(ERF, 1-(z-1)*Rdn)

where Rdn is a slope parameter between 0.2 and 0.8, and ERF is the minimum decrease required. The source should never allowed to transmit at a rate higher than the fair share. The distinguishing feature of this scheme is that it is oscillation-free in steady state.

4.2.5 ER Based on Bandwidth Demand Estimate Algorithm

The switch calculates the Mean Allowed Cell rate (MACR) basing on a running exponential average of the ACR value from each VC's forward RM-cells as:

MACR = MACR + (ACR - MACR) * AVF

where AVF (ACR Variation Factor) is set to 1/16.

If the load factor is less than 1, the left-over bandwidth is reallocated according to:

MACR = MACR + MAIR

where MAIR is the MACR Additive Increase Rate.

The ER value is computed as:

ER = MACR * MRF

where MRF is the MACR Reduction Factor if congestion is detected,

ER = MACR if no congestion is detected. The congestion condition is detected by observing that the queue derivative is positive.

CHAPTER 5

ANALYSIS OF ERICA RATE-BASED CONGESTION CONTROL SCHEME

The ERICA (Explicit Rate Indication for Congestion Avoidance) algorithm is concerned with the fair and efficient allocation of the available bandwidth to all contending sources. Like any dynamic resource algorithm, it requires monitoring the available capacity and the current demand on the ~~resources. Here, the key "reso~~urce" is the available bandwidth at a queueing point (input or output port). In most switches, output buffering is used, which means that most of the queueing happens at the outport ports. Thus, ERICA algorithm is applied to each output port (or hike).

5.1 The Basic Algorithm

The switch periodically monitors the load on each link and determines a load factor, z, the available capacity, and the number of currently active VCs (N).

The load factor is calculated as the ratio of the measured input rate at the port to the target capacity of the output link.

$$z = \frac{\text{ABR Input Rate}}{\text{ABR Capacity}}$$

where,

ABR Capacity = Target Utilization (U) x Link Bandwidth

The Input Rate is measured over an interval called the switch averaging interval. The above steps are executed at the end of the switch averaging interval.

Target utilization (U) is a parameter which is set to a fraction (close to, but less than 100%) of the available capacity. Typical values of target utilization are 0.9 and 0.95.

The load factor, z, is an indicator of the congestion level of the link. High overload values are undesirable because they indicate excessive congestion; so are low overload values which indicate link underutilization. The optimal operating point is at an overload value equal to one. The goal of the switch is to maintain the network at unit overload.

The fair share of each VC, Fair Share, is also computed as follows

$$\text{Fair Share} = \frac{\text{ABR Capacity}}{\text{Number of Active Sources}}$$

The switch allows each source sending at a rate below the Fair Share to rise to Fair Share every time it sends a feedback to the source. If the source does not use all of its Fair Share, then the switch fairly allocates the remaining capacity to the sources which can use it. For this purpose, the switch, the switch calculates the quantity:

$$\text{VCShare} = \frac{\text{CCR}}{z}$$

If all VCs changed their rate to their VCShare values then, in the next cycle, the switch would experience unit overload (z equals one).

Hence VCShare aims at bringing the system to an efficient operating point, which may not necessarily be fair, and Fair Share allocation aims at ensuring fairness, possibly leading to overload (inefficient operation). A combination of these two quantities is used to rapidly reach optimal operation as follows.

$$\text{ER Calculated} = \text{Max (FairShare, VCShare)}$$

A complete flow chart of the algorithm is presented in Fig. 1. The flow chart shows step to be taken on three possible events: at the end of an averaging interval, on receiving a cell (data or RM), and on receiving a backward RM cell. These steps have been numbered.

5.2 Achieving Max-Min Fairness

Assuming that the measurements do not suffer from high variance, the above algorithm is sufficient to converge to efficient operation in all cases and to the max-min fair allocations in most cases. The convergence from transient conditions to the desired operating point is rapid, often taking less than a round trip time.

This happens if all of the following three conditions are met:

1. The load factor z becomes one
2. There are some sources which are bottlenecked elsewhere upstream.
3. CCR for all remaining sources is greater than the Fair Share.

To achieve max-min fairness, the basic ERICA algorithm is extended by remembering the highest allocation made during one averaging interval and ensuring that all eligible sources can also get this high allocation.

Basically, for $z > 1+\delta$, where δ is a small fraction, we use the basic ERICA algorithm and alocate the source Max (FairShare, VCShare). But, for $z <= 1+\delta$, we attempt to make all the rate allocations equal. We calculate the ER as Max (FairShare, VCShare, MaxAllocPrevious).

The key point is that the VCShare is only used to achieve efficiency. The fairness can be achieved only by giving the contending sources equal rates. The system is considered to be in a state of overload when its load factor, z, is greater than $1+\delta$.

5.3 Fairshare First to Avoid Transient Overloads

The inter-RM cell time determines how frequently a source receives feedback. It is also a factor in determining the transient response time when load conditions change. With the basic ERICA scheme, it is possible that a source which receives feedback first can keep getting rate increase indications, purely because it sends more RM cells before competing sources can receive feedback.

The problem arises when the Backward RM (BRM) cells from different sources arrive asynchronously at the switch. Consider a LAN configuration of two sources (A and B), initially sending at low rates. When the BRM arrives, the switch calculates the feedback for the current load. The transient overload experienced at the switch may still be below unity, and the ACR of source A is increased further (BRMs for source A are available since source A sends more RM cells at higher rates). This effect is observed as an undesirable spike in the ACR graphs and sudden queue spikes when the source B gets its fair share.

The problem can be solved by incorporating the following changes to the ERICA algorithm. When the calculated ER is greater than the fair share value, and the source is increasing from CCR below FairShare, we limit its increase to FairShare. Alternatively, the switch could decide not to give new feedback to this source for one measurement interval. This is useful in LANs where the round trip time is shorter that the inter-RM cell gap and the switch measurement interval. The following commutation is added to the switch algorithm.

After "ER Calculated is computed:

```
IF ((CCR < FairShare) AND (ER Calculated ≥ FairShare)) THEN
    ER Calculated = FairShare
```

We can also disable feedback to this source for one measurement interval.

"ER in RM Cell" is then computed as before.

5.4 Forward CCR used for Reverse Direction Feedback

The only requirement for each switch is to provide its feedback to the sources. This can also be achieved if it indicates the feedback in the reverse path of the RM cell. The backward going RM (BRM) cell takes less time to reach the source than the forward going RM (ARM) cell which has to reach the destination first. Thus, the system responds faster to changes in the load level. However, the CCR carried by the BRM cell no longer reflects the load level in the system. To maintain the most current CCR value, the switch copies the CCR field from ARM cells, and uses this information to compute the ER value to be inserted in the BRM cells. This ensures that the latest CCR information is used in the ER calculation and that the feedback path is as short as possible. Figure 2 shows that the first RM cell carries (in its backward path), the feedback calculated from the information in the most recent FRM cell. The CCR table update and read operations still preserve the O(!) time complexity of the algorithm.

5.5 Feedback in a Switch Interval

The switch measures the overload, the number of active sources and the ABR capacity periodically (at the end of every switch averaging interval). The source also sends RM cells periodically (once every Nrm cells). These RM cells may contain different rates in their CCR fields. If the switch encounters more than one RM cell from the same VC during the same switch interval, then it uses the same value of overload for computing feedback in both cases.

ERICA adopts an approach, where the source and the switch intervals need not be corrected. The switch provides only one feedback value during each switch interval irrespective of the number of RM cells it encounters. The switch calculates the ER only once per interval, and the ER value obtained is stored. It inserts the same ER value in all the RM cells it sees during this interval. The source and switch intervals are completely independent. The source independently decides the inter-RM cell distance, thus determining the frequency of feedback. In Fig.3 the switch interval is greater than the RM cell distance. The ER calculated in the interval marked Load Measurement Interval is maintained in a Table and set in all the RM cells passing through the switch during the next interval.

5.6 Per-VC CCR Measurement Option

The CCR of a source is obtained from the CCR field of the forward going RM cell. The latest CCR value is used in the ERICA computation. It is assumed that the CCR is correlated with load factor measured. When the CCR is low, the frequency of forward RM cells becomes very low. Hence, the switch may not have a new CCR estimate though a number of averaging intervals have elasped. Moreover, the CCR value may not be an accurate measure of the rate of the VC if the VC is bottlenecked at the source, and is not able to use its ACR allocation. Note that if a VC is bottlenecked on another link, the CCR is set to the bottleneck allocation within one round-trip.

A possible solution to the problems of inaccurate CCR estimates is to measure the CCR of every VC during the same averaging interval as the load factor. This requires the switch to count the number of cells received per VC during every averaging interval and update the estimate as follows:

```
At the end of an switch averaging interval :

    FOR ALL VCs DO
        CCR [VC] = NumberOfCells[VC]/IntervalLength
        NumberOfCells[VC] = 0
    END
When a cell is received :
    NumberOfCells[VC] = NumberOfCells[VC] + 1
Initialization :
    FOR ALL VCs DO NumberOfCells[VC] = 0
When an FRM cell is received, do not copy CCR field from FRM into CCR[VC].
```

The effect of the per VC CCR measurement can be explained as follows. The basic ERICA uses the following formula :

ER Calculated = Mx (FairShare, VCShare)

The measured CCR estimate is always less than or equal to the estimate obtained from the RM cell CCR field. If the other quantities remain constant, the term "VCShare" decreases. Thus the ER calculated will decrease whenever the first term dominates. This change results in a more conservative feedback, and hence shorter queues at the switches.

5.7 ABR Operation with VBR and CBR in the Background

Normally, ATM links are used by constant bit rate (CBR) and variability bit rate (VBR) traffic along with ABR traffic. In fact, CBR and VBR have a higher priority. For such links, we need to measure the CBR and VBR usage along with the input rate. The ABR capacity is then calculated as follows :

ABR Capacity = Target Utilization x Link Bandwidth - VBR Usage - CBR Usage

The rest of ERICA algorithm remains unchanged. The target utilization is applied to the entire link bandwidth and not to the left over capacity. That is,

ABR Capacity ≠ Target Utilization x {Link Bandwidth - VBR Usage - CBR Usage}

There are two implications of this choice. First,)1-Target Utilisation) x (link bandwidth) is available to drain the queues, which is much more that what would be available otherwise. Second, the sum of VBR and CBR usage must be less than (target utilisation) x (link bandwidth).

5.8 Bi-directional Counting of Bursty Sources

A bursty source sends data in bursts during its active periods, and remains idle during other periods. It is possible that the BRM cell of a bursty source could be traveling in the reverse direction, but no cells of this source are travailing in the forward direction. A possible enhancement to the counting algorithm is to also count a source as active whenever a BRM of this source is encountered in the reverse direction. This is referred as "bi-directional counting of active VCs".

One problem with this technique is that the reverse queues may be small and the feedback may be given before the FairShare is updated, taking into consideration the existence of the new source.

We could also reset the CCR of such a source to zero after updating the FaitShare value, so that the source is not allocated more than the FairShare value. The motivation behind this strategy is that the source may be idle, but its CCR is unchanged because no new FRMs are encountered. The setting of CCR to zero is a conservative strategy which avoids large queues due to bursty or ACR retaining sources. Only drawback of this strategy is that in certain configurations, the link may not be fully utilized if the entire traffic bursty.

5.9 Averaging of the Number of Sources

A technique to overcome the problem of underestimating the number of active sources is to use exponential averaging to decay the contribution of each VC to the number of active source count.

Flow charts of Figure 4 and 5 show this technique.

The DecayFactor used in decaying the contribution of each VC is a value between zero and one, and is usually selected to be a large fraction. Setting the DecayFactor to a smaller fraction makes the scheme adapt faster to sources which become idle, but makes the scheme more sensitive to the averaging interval length.

5.10 Boundary Cases

Two boundary conditions are introduced in the calculations at the end of the averaging interval. First, the estimated number of active sources should never be less than one. If the calculated number of sources is less than one, the variable is set to one. Second, the load factor becomes infinity when the ABR capacity is measured to be zero, and the load factor becomes zero when the input rate is measured to be zero. The corresponding allocations are described in Table 1.

Table 1 : Boundary Cases

ABR Capacity	Input Rate	Overload	Fairshare	CCR/Overload	Feedback
Zero	Non-zero	Infinity	Zero	Zero	Zero
Non-zero	Zero	Infinity	C/N	Zero	C/N
Non-zero	Non-zero	I/C	C/N	CCR*C/I	Max (CCR*C/I, C/N)
Zero	Zero	Infinity	Zero	Zero	Zero

5.11 Averaging of the Load Factor

In cases where no input cells are seen in an interval, or when the ABR capacity changes suddenly (possible due to a VBR source going away), the overload measured in successive intervals may be considerably different. This leads to considerably different feedbacks in successive intervals. An optional enhancement to smoothen this variance is by averaging the load factor. This effective increases the length of the averaging interval over which the load factor is measured. One way to accomplish this is shown in the flow chart of Figure 5.

The method described above has the following drawbacks. First, the average is reset everytime z becomes infinity. The entire history accumulated in the average prior to the interval where the load is to be infinity is lost.

The second problem with this method is that the exponential average does not give a good indication of the average value of quantities which are not additive. In our case, the load factor is not an additive quantity. However, the number of ABR cells received or output is additive.

To average load factor, we need to average the input rate (numerator) and the ABR capacity (denominator) separately. However, input rate and the ABR capacity are themselves ratios of cells over time. The input rate is the ratio of number of cells input and the averaging interval. If the input rates are x1/T1, x2.T2, ..., xn/Tn, the average input rate is ((x1 + x2 + ... + xn)/(T1 + T2 + ... + Tn)/n). Here, xi's are the number of ABR cells input in averaging interval i of length Ti. Similarly the average ABR capacity is (y1 + y2 + ... + yn)/n)/((T1 + T2

+ ... + Tn)/n). Here, yi's are the maximum number of ABR cells that can be output in averaging interval i of length Ti.

The load factor is the ratio of these two averages.

Exponential averaging is an extension of arithmetic averaging used above. Hence, the averaging like (x1 + x2 + ... + xn) can be replaced by the exponential average of the variable xi.

The flow chart of Figure 5 describes this averaging method.

5.12 Time and Count Based Averaging

The load factor, available ABR capacity and the number of active sources need to be measured periodically. The averaging interval can be set as the time required ti receive a fixed number of ABR cells (M) at the switch in the forward direction. While this definition is sufficient to correctly measure the load factor and the ABR capacity at the switch, it is not sufficiently to measure the number of active VCs (N) or the CCR per VC accurately.

An alternative way of averaging the quantities is by fixed time interval, T. This ensures that any source sending at a rate greater than (one cell/T) will be encountered in the averaging interval.

One way of combining these two kinds of intervals is to use the minimum of the fixed-cell interval and fixed-time interval. Another strategy for overcoming this limitation could be to measure N and per-VC CCR over a fixed-time interval; and the capacity and load factor over the minimum of the fixed-cell and fixed-time interval.

5.13 Selection of ERICA Parameters

Most congestion control schemes provide the network administrator with a number of parameters that can be set to adopt the behaviour of the schemes to their needs. A good scheme must provide a small number of parameters that offer the desired level of control. These parameters should be relatively insensitive to minor changes in network characteristics.

ERICA provides a few parameters which are easy to set because the tradeoffs between their values are well understood. Here, two parameters are provided : the Target Utilization (U) and the Switch Measurement Interval.

The target Utilization determines the link utilization during steady state conditions. If the input rate is greater than Target Utilization x Link Capacity, then the switch asks sources to decrease their rates to bring the total input rate to the desired fraction. If queues are present in the switch due to transient overloads, then (1-U) x Link Capacity is used to drain the queues. The network administrator is free to set the values of Target Utilization as desired.

ERICA measure the required quantities over an averaging interval and uses the measured quantities to calculate the feedback in the next averaging interval.

5.14 Two Class Scheduling : ABR and VBR

Since the switches provide multiple classes of service, they maintain multiple queues. The key question is how cells in these different queues are serviced. For example, in the case of a simple two class *VBR and ABR) system, an implementator could decide to give VBR a maximum of 90% and ABR a minimum of 10% bandwidth. If total ABR load is only 20%, ABR gets the remaining 80%. On the other hand if VBR input rate is 110% and ABR input rate is 15%, VBR gets only 90% and ABR gets 10%. If VBR and ABR are 110% and 5%, VBR gets 95% and ABR gets 5%.

Consider the two categories ABR and VBR. The VBR service class is characterised by PCR and SCR parameters which the network must provide to the VBR class. The ABR service class on the other hand is characterised by MCR. The network only guarantees a minimum bandwidth ofMCR to the ABR class. Any other available bandwidth is also allocated to this class. Since VBR applications are delay sensitive while ABR applications are not, VBR can be considered to be a higher priority class than ABR.

Let vfrac and afrac be the fractions of the total link capacity allocated to VBR and ABR respectively. If VBR and ABR are the only two supported service categories, then we can assume without loss of generality that

vfrac + afrac = 1

If both classes have cells to send at all times, then for every n cells (for large enough n), n*afrac ABR cells and n*vfrac VBR cells must be scheduled.

The scheduling is implemented by a policy described below. The scheduler keeps track of the relative proportions of bandwidth currently used by each class by maintaining credit variables acredit and vcredit.

- The class with higher credit value is determined eligible to be scheduled. If credits are equal, then VBR is eligible to be scheduled.
- If the eligible class has cells in its buffer, a cell from this class is scheduled, 1 is subtracted from the credit of the class. If the eligible class cannot be scheduled, then the other class is scheduled if possible but 1 is not subtracted from any credit value.
- The credit value of each class is incremented by the corresponding fraction.

The flow chart of Fig. shows the above algorithm. The pseudocode is given in the appendix A.

5.15 Possible Modification to ERICA

ERICA depends upon the measurement of metrics like overload factor, and the number of active ABR sources. If there is a high error in the measurement, and the target utilization is set to very high values, ERICA may diverge.

One simple enhancement of that can be done to ERICA is to have a queue threshold, and reduce the target utilization if the queue is greater than the threshold. Once the target utilization is low, the queues are drained out quickly. Hence, this enhancement could maintain high utilization when the queues are small, and drains out queues quickly when they become high.

ERICA achieves high utilization in the steady state, but utilization is limited by the target utilization parameter. The way to get 100% utilization in steady state, and quick drainage of queues is to vary the target ABR rate dynamically. Then the target rate would be function of queue length, link rate and VBR rate.

One feature of ABR is that its capacity varies dynamically, due to the presence of higher priority classes (CBR and VBR). Hence, if the higher priority classes are absent for a short interval (which may be smaller than the feedback delay), the remaining capacity is not utilized. Hence, it could be useful to have a bucket full of ABR cells.

CONCLUSIONS

Congestion control for ATM networks enompasses a number of interrelated elements operating over different levels and time scales. This report introduces the concepts in congestion control for ATM networks and evolution of rate-based congestion control schemes for ABR service has been traced. Some rate based congestion control schemes are also described.

We have analysed a congestion avoidance scheme called (ERICA) for data traffic in ATM networks. We see that the ERICA scheme achieves both efficiency and fairness, and exhibits a fast transient response. The development of the scheme was also traced and the approaches it uses to achieve its objectives were highlighted. Several design and implementation aspects of the scheme were examined and its performance was discussed. In addition several possible enhancements to the above scheme have also been discussed. The pseudocode for the ERICA algorithm is given in Appendix.

BIBLIOGRAPHY

1. K. Sriram and W. Whitt, Characterizing superposition arrival processes in packet multiplexers for voice and data, *IEEE, Trans, Selected Areas Commun.* 4(6)(1986) 833-846.

2. R. Krishanan, A comparison of per-VC queuing and explicit rate-setting mechanism, in: ATM Forum Technical Committee Meeting, Ottawa, 1994.

3. F. Bonomi, K.W. Fendick and K. Meier-Hellstern, A comparative study of EPCRA compatible schemes for the support of fair ABR Forum Technical Committee Meeting, Ottawa, 1994.

4. H.T. Kung and R. Morris, Credit-based flow control for ATM networks, *IEEE Network* 9(2) (1995) 40-48.

5. K.K. Ramakrishnan and P. Jain, A binary feedback scheme for congestion avoidance in computer networks, *ACM Trans, Comput. Systems* 8(2) (1990) 158-181.

6. R. Jain, Congestion control in computer networks: Issues and trends, *IEEE Network Mag.* (May 1990) 24-30.

7. R.E. Boyer and D.P. Tranchier, A reservation principle with applications to the ATM traffic control, *Comput, Networks ISDN Systems*, 24(1992) 321-334.

8. K.K. Ramakrishnan and J. Zavgren, Preliminary simulation results of hop-by-hop/VC flow control and early packet discard, AF-TM 94-0231, March 1994.

9. W. Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM* (Prentice HAll, Englewood Cliffs, NJ, 1995) 581.

10. P. Newman, Traffic Management for ATM Local Area Networks", *IEEE Commun. Mag.*, Vol.32 no. 32, no. 8, Aug. 1994, pp. 44-50.

11. D. Bartsekas and R. Gallager, Data Networks, Prentice Hall, 2nd Edition, (1987).

12. K.W. Fendick and M.A. Rodrigues, Asymptotic Analysis of Adaptive Rate Control for

Diverse Sources with Delayed Feedback, *IEEE Trans. Info. Theory*, Vol.40 No.4, Nov. 1994, pp. 2008-2025.

13. S.J. Golestani, A self-clocked fair queuing scheme for broadband applications, *Proc. IEEE INFOCOM*, 1994, pp. 636-646.

14. N. Yin and M. Hluchyi. On Closed-Loop rate controls for ATM cell relay networks, *IEEE INFOCOM'94*, June 1994.

15. P. Newman, Backward explicit congestion notification for ATM local area networks, *IEEE GLOBECOM'93*, pp. 719-723, December 1993.

16. R.Jain, Myths about congestion management in high-speed networks, *in Information Network andData Communications, IV* (M. Tienari and D. Khakhar, eds.), pp. 55-70, Elsevier Science Publishers B.V. (North Holland), 1992.

17. A. Berger, F. Bonomi, K. Fendick, and J. Swenson, Control of multiplexed connections using backward congestion control, *ATM Forum Contribution 94-064*, January 1994.

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

19. Tanenbaum, Computer networks, 3rd Edition, PHI.

20. Hiroyuki Ohsaki, Masayuki Murata, Hinroshi Suzuki, Chinatns Jkedr and Hideo Miyahara, Rate based congestion control for ATM networks, *ACM SIGCOMN, Computer Comm. Review*.

21. A. Arulambalam and Xizogiang Chen, Allocating Fair Rtaes for Available Bit rate service in ATM Networks, *IEEE Comm. Mag.*, Nov. 1996.

22. R. Jain, S. Kalyanaruma, S. Fahmy and R. Goyal, Source behaviour for ATM ABR traffic management: An explanation, *IEEE Comm. Mag.*, Nov. 1996.

23. M. Decina, U. Trecordi, Traffic management and congestion control for ATM networks., *IEEE Network Mag.*, Sept. 1992.

24. A.E. Eckberg, B-JSDN/ATM traffic and congestion control., *IEEE Network Mag.*, Sept. 1992.

APPENDIX

SWITCH PSEUDOCODE

Notes :

- All rates are in the units of cell/s
- The following pseudo-code assumes a simple fixed-time averaging interval. Extension to a cells and time averaging interval is trivial

We use the folowing identifying names of flow charts :

Flow Chart 1 : Flow Chart of the Basic ERICS Algorithm. Figure 2

Flow Chart 2 : Flow Chart for Achieving Max-Min Fairness. Figure 3

Flow Chart 3 : Flow Chart for Bi-Directional Counting. Figure 6

Flow Chart 4 : Flow Chart of Averaging Number of Active Sources (Part 1 of 2). Figure 7

Flow Chart 5 : Flow Chart of Averaging Number of Active Sources (Part 2 of 2). Figure 8

Flow Chart 6 : Flow Chart of Averaging Load Factor (Method 1). Figure 9

Flow Chart 7 : Flow Chart of Averaging Load Factor (Method 2). Figure 10

Flow Chart 6 : Flow Chart of 2-class Scheduling. Figure 11

Explanation of some of the variables used are given in the following pages.

Name	Explanation	Flow Chart/Figure
ABR_Cell_Count	Number of ABR input cells in the current interval	Flow charts 1 and 7 (step 2)
Contribution[VC]	Contribution of the VC towards the count of the number of active sources	Flow charts 4 and 5
Seen_VC_In_This_Interval[VC] Seen_VC_In_Last_Interval[VC]	A bit which is set when a VC is seen in the current (last) interval	Flow charts 1,3 and 5
Number_Of_Cells[VC]	Used in Per VC CCR option to count number of cells from each VC in the current interval	
Max_Alloc_Previous	Max rate allocation in previous interval	Flow chart 2
Max_Alloc_Current	Max rate allocation in current interval	Flow chart 2
VBR_Credit ABR_Credit	Credit variables used in scheduling	Flow chart 8
Seen_BRM_Cell_In_This_Interval[VC]	A BRM from the source has been seen (and feedback given) in this interval. Do not give new feedback	Figure 5
Last_Allocated_ER	Unique ER feedback to the source in the current interval	Figure 5
Decay_Factor	Factor Used in Averaging the Number of Active Sources $0 < \text{Decay\_Factor} \leq 1$	Flow Charts 4 and 5

Initialization:

```
(* ABR Capacity and Target Utilization *)
IF (Queue_Control_Option) THEN
   Target_Utilization ←1
END (* IF *)
ABR_Capacity_In_cps ←Target_Utilization×Link_Bandwidth − VBR_and_CBR_Capacity

(* Count of Number of VCs, Cells *)
FOR ALL VCs DO
   Contribution[VC] ←0
   Seen_VC_In_This_Interval[VC] ←0
   Seen_BRM_Cell_In_This_Interval[VC] ←0
END (* FOR *)
```

```
ABR_Cell_Count ←ABR_Capacity_In_cps × Averaging_Interval
Number_Active_VCs_In_This_Interval ←Total Number of Setup VCs
Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_This_Interval

(* Fairshare and Load Factor variables *) Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last
Max_Alloc_Previous ←0
Max_Alloc_Current ←Fair_Share
Load_Factor ←ABR_Capacity_In_cps/ FairShare

(* Per VC CCR Option Variables *)
IF (Per_VC_CCR_Option) THEN
   FOR ALL VCs DO
     Number_Of_Cells[VC] ←0
   END (* FOR *)
END (* IF *)

(* 2-class Scheduling variables *)
VBR_Fraction, ABR_Fraction ←preassigned bandwidth fractions
VBR_Credit ←VBR_Fraction
ABR_Credit ←ABR_Fraction
```

A cell of "VC" is received in the forward direction:

```
IF (Averaging_VCs_Option) THEN
   IF (Contribution[VC] < 1) THEN (* VC inactive in current interval *)
     Number_Active_VCs_In_This_Interval ←
          Number_Active_VCs_In_This_Interval − Contribution[VC] + 1
     IF ((Immediate_Fairshare_Update_Option) AND (Contribution[VC] < Decay_Factor)) THEN
       Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval
          − (Contribution[VC] / Decay_Factor) + 1
       Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
     END (* IF *)
     Contribution[VC] ←1
   END (* IF *)
ELSE
   IF (NOT(Seen_VC_In_This_Interval[VC])) THEN
     Seen_VC_In_This_Interval[VC] ←1
   END (* IF *)
   IF ((Immediate_Fair_Share_Option) AND (NOT(Seen_VC_In_Last_Interval[VC]))) THEN
     Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval + 1
     Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
     Seen_VC_In_Last_Interval[VC] ←1
   END (* IF *)
END (* IF *)
ABR_Cell_Count ←ABR_Cell_Count + 1
```

```
IF (Per_VC_CCR_Option) THEN
  Number_Of_Cells[VC] ←Number_Of_Cells[VC] + 1
END (* IF *)
```

Averaging interval timer expires:

```
IF (NOT(Averaging_VCs_Option)) THEN
  Number_Active_VCs_In_Last_Interval ←Max ($\sum$ Seen_VC_In_This_Interval, 1)
  Number_Active_VCs_In_This_Interval ←0
  FOR ALL VCs DO
    Seen_VC_In_Last_Interval[VC] ←Seen_VC_In_This_Interval[VC]
  END (* FOR *)
ELSE
  Number_Active_VCs_In_Last_Interval ←Max(Number_Active_VCs_In_This_Interval,1)
  Number_Active_VCs_In_This_Interval ←0
  FOR ALL VCs DO
    Contribution[VC] ←Contribution[VC] × Decay_Factor
    Number_Active_VCs_In_This_Interval ←Number_Active_VCs_In_This_Interval + Contribution[VC]
  END (* FOR *)
END (* IF *)

IF (Exponential_Averaging_Of_Load_Method_2_Option) THEN
  ABR_Capacity_In_Cells ←Max(Target_Utilization×Link_Bandwidth× Averaging_Interval
    − VBR_and_CBR_Cell_Count, 0)
  Avg_ABR_Capacity_In_Cells ←$(1-\alpha)\times$ Avg_ABR_Capacity_In_Cells + $\alpha\times$ ABR_Capacity_In_Cells
  Avg_Averaging_Interval ←$(1-\alpha)\times$ Avg_Averaging_Interval + $\alpha\times$Averaging_Interval
  Avg_ABR_Cell_Count ←$(1-\alpha)\times$Avg_ABR_Cell_Count + $\alpha\times$ABR_Cell_Count
  ABR_Input_Rate ←Avg_ABR_Cell_Count / Avg_Averaging_Interval
  ABR_Capacity_In_cps ←Avg_ABR_Capacity_In_Cells / Avg_Averaging_Interval
ELSE
  VBR_and_CBR_Cell_Rate ←VBR_and_CBR_Cell_Count / Averaging_Interval
  ABR_Capacity_In_cps ←
    Max(Target_Utilization×Link_Bandwidth − VBR_and_CBR_Cell_Rate, 0)
  ABR_Input_Rate ←ABR_Cell_Count / Averaging_Interval
END (* IF *)

IF (Queue_Control_Option) THEN
  Target_Queue_Length ←Target_Time_To_Empty_Queue × ABR_Capacity_In_cps
  Queue_Control_Factor ←Fn(Current_Queue_Length)
  ABR_Capacity_In_cps ←Queue_Control_Factor × ABR_Capacity_In_cps
END (* IF *)

IF (Exponential_Averaging_Of_Load_Method_1_Option) THEN
  IF (ABR_Capacity_In_cps $\leq 0$) THEN
    Load_Factor ←Infinity
  ELSE
    IF (Load_Factor = Infinity) THEN
      Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
    ELSE
      Load_Factor ←(1−$\alpha$) × Load_Factor + $\alpha$ × ABR_Input_Rate / ABR_Capacity_In_cps
    END (* IF *)
  END (* IF *)
ELSE IF (Exponential_Averaging_Of_Load_Method_2_Option) THEN
  IF (ABR_Capacity_In_cps $\leq$ 0) THEN
    Load_Factor ←Infinity
  ELSE
    Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
  END (* IF *)
ELSE (* No exponential averaging *)
  IF (ABR_Capacity_In_cps $\leq$ 0) THEN
    Load_Factor ←Infinity
  ELSE
    Load_Factor ←ABR_Input_Rate / ABR_Capacity_In_cps
  END (* IF *)
END (* IF *)

Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
Max_Alloc_Previous ←Max_Alloc_Current
Max_Alloc_Current ←Fair_Share
FOR ALL VCs DO
  Seen_VC_In_This_Interval[VC] ←0
  Seen_BRM_Cell_In_This_Interval[VC] ←0
END (* FOR *)
ABR_Cell_Count ←0
IF (Per_VC_CCR_Option) THEN
  FOR ALL VCs DO
    CCR[VC] ←Number_Of_Cells[VC]/Averaging_Interval
    Number_Of_Cells[VC] ←0
  END (* FOR *)
END (* IF *)
VBR_and_CBR_Cell_Count ←0
Restart Averaging_Interval Timer
```

A Forward RM (FRM) cell of "VC" is received:

```
IF (NOT(Per_VC_CCR_Option)) THEN
  CCR[VC] ←CCR_In_FRM_Cell
END (* IF *)
```

A Backward RM (BRM) cell of "VC" is received:

```
IF (Averaging_VCs_Option) THEN
   IF (Contribution[VC] < 1) THEN (* VC inactive in current interval *)
      Number_Active_VCs_In_This_Interval ←
         Number_Active_VCs_In_This_Interval − Contribution[VC] + 1
      IF ((Immediate_Fairshare_Update_Option) AND (Contribution[VC] < Decay_Factor)) THEN
         Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval
            − (Contribution[VC] / Decay_Factor) + 1
         Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
      END (* IF (Immediate ...) *)
      Contribution[VC] ←1
   END (* IF (Contribution ... ) *)
ELSE (* NOT (Averaging_VCs_Option) *)
   IF (NOT(Seen_VC_In_This_Interval[VC])) THEN
      Seen_VC_In_This_Interval[VC] ←1
   END (* IF *)
   IF ((Immediate_Fair_Share_Option) AND (NOT(Seen_VC_In_Last_Interval[VC]))) THEN
      Number_Active_VCs_In_Last_Interval ←Number_Active_VCs_In_Last_Interval + 1
      Fair_Share ←ABR_Capacity_In_cps / Number_Active_VCs_In_Last_Interval
      Seen_VC_In_Last_Interval[VC] ←1
   END (* IF ((Immediate ..)) *)
END (* IF-THEN-ELSE (Averaging_VCs_Option) *)

IF (Seen_BRM_Cell_In_This_Interval[VC]) THEN
   ER_Calculated ←Last_Allocated_ER[VC]
ELSE
   VC_Share[VC] ←CCR[VC] / Load_Factor
   (* Max-Min Fairness Algorithm *)
   IF (Load_Factor > 1 + δ) THEN
      ER_Calculated ←Max (Fair_Share, VC_Share)
   ELSE
      ER_Calculated ←Max (Fair_Share, VC_Share, Max_Alloc_Previous)
   END (* IF *)
   Max_Alloc_Current ←Max (Max_Alloc_Current, ER_Calculated)
   (* Avoid Unnecessary Transient Overloads *)
   IF ((CCR[VC] < Fair_Share) AND (ER_Calculated ≥ Fair_Share)) THEN
      ER_Calculated ←Fair_Share
      (* Optionally Disable Feedback To This VC For An Averaging Interval *)
   END (* IF *)
   ER_Calculated ←Min(ER_Calculated, ABR_Capacity_In_cps)
```

```
  (* Ensure One Feedback Per Switch Averaging Interval *)
  Last_Allocated_ER[VC] ←ER_Calculated
  Seen_BRM_Cell_In_This_Interval[VC] ←1
END (* IF *)

(* Give Feedback In BRM Cell *)
ER_In_BRM_Cell ←Min (ER_in_BRM_Cell, ER_Calculated)
```

At each cell slot time (two-class scheduling):

```
IF (VBR_Credit ≥ ABR_Credit) THEN
   IF (VBR Queue is Non-empty) THEN
      Schedule VBR Cell
      IF (ABR Queue is Non-empty) THEN
         VBR_Credit ←VBR_Credit − 1
      END (* IF *)
      VBR_Credit ←VBR_Credit + VBR_Fraction
      ABR_Credit ←ABR_Credit + ABR_Fraction
   ELSE IF (ABR Queue is Non-empty) THEN
      Schedule ABR Cell
   END (* IF-THEN-ELSE (VBR Queue is Non-empty) *)
ELSE (* NOT (VBR_Credit ≥ ABR_Credit) *)
   IF (ABR Queue is Non-empty) THEN
      Schedule ABR Cell
      IF (VBR Queue is Non-empty) THEN
         ABR_Credit ←ABR_Credit − 1
      END (* IF *)
      ABR_Credit ←ABR_Credit + ABR_Fraction
      VBR_Credit ←VBR_Credit + VBR_Fraction
   ELSE IF (VBR Queue is Non-empty) THEN
      Schedule VBR Cell
   END (* IF-THEN-ELSE (ABR Queue is Non-empty) *)
END (* IF-THEN-ELSE (VBR_Credit ≥ ABR_Credit) *)
```

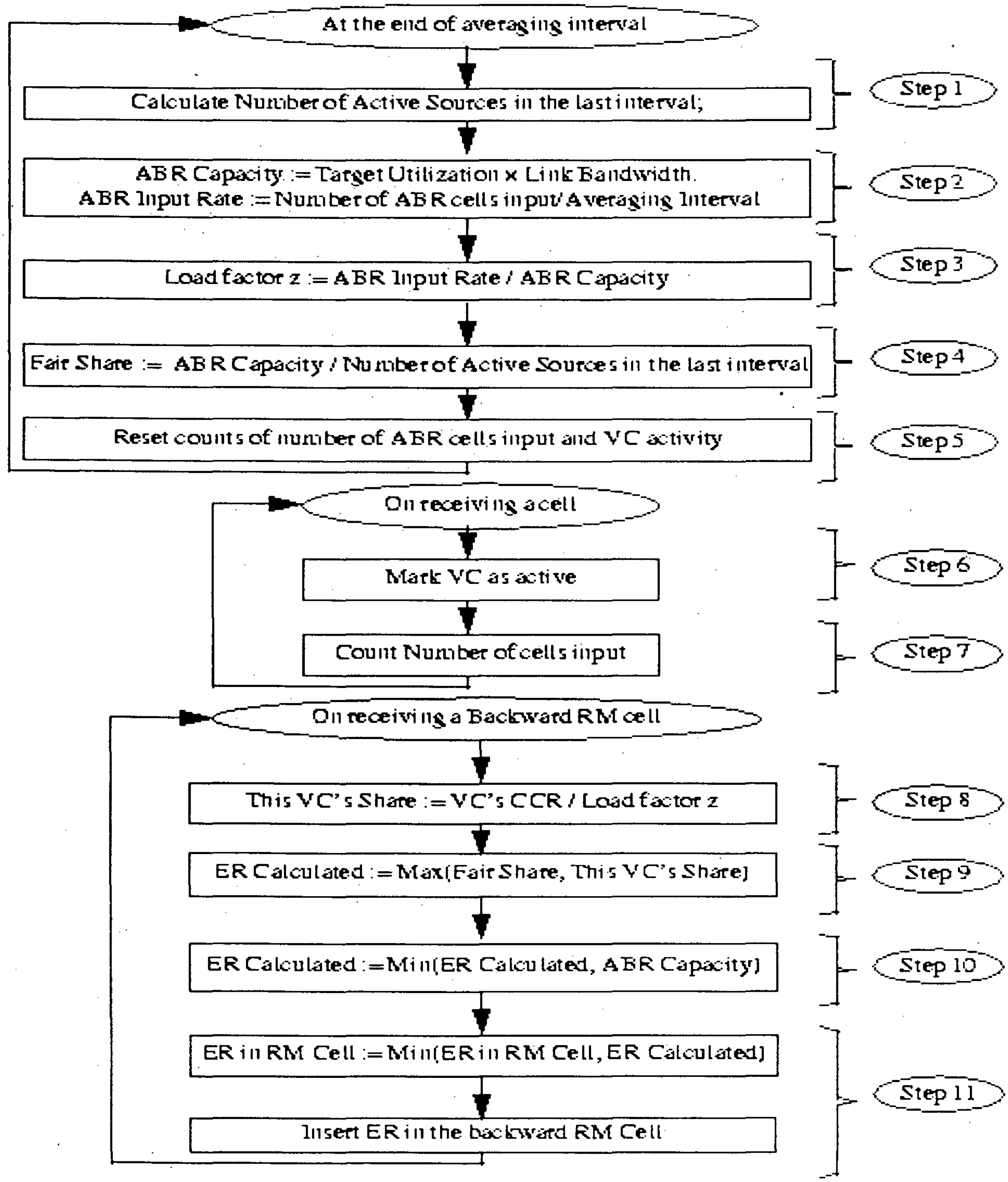
At the end of averaging interval
Calculate Number of Active Sources in the last interval;
ABR Capacity := Target Utilization x Link Bandwidth.
ABR Input Rate := Number of ABR cells input/Averaging Interval
Load factor z := ABR Input Rate / ABR Capacity
Fair Share := ABR Capacity / Number of Active Sources in the last interval
Reset counts of number of ABR cells input and VC activity
On receiving a cell
Mark VC as active
Count Number of cells input
On receiving a Backward RM cell
This VC's Share := VC's CCR / Load factor z
ER Calculated := Max(Fair Share, This VC's Share)
ER Calculated := Min(ER Calculated, ABR Capacity)
ER in RM Cell := Min(ER in RM Cell, ER Calculated)
Insert ER in the backward RM Cell
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7
Step 8
Step 9
Step 10
Step 11

Figure 2 Flow Chart of the Basic ERICA Algorithm

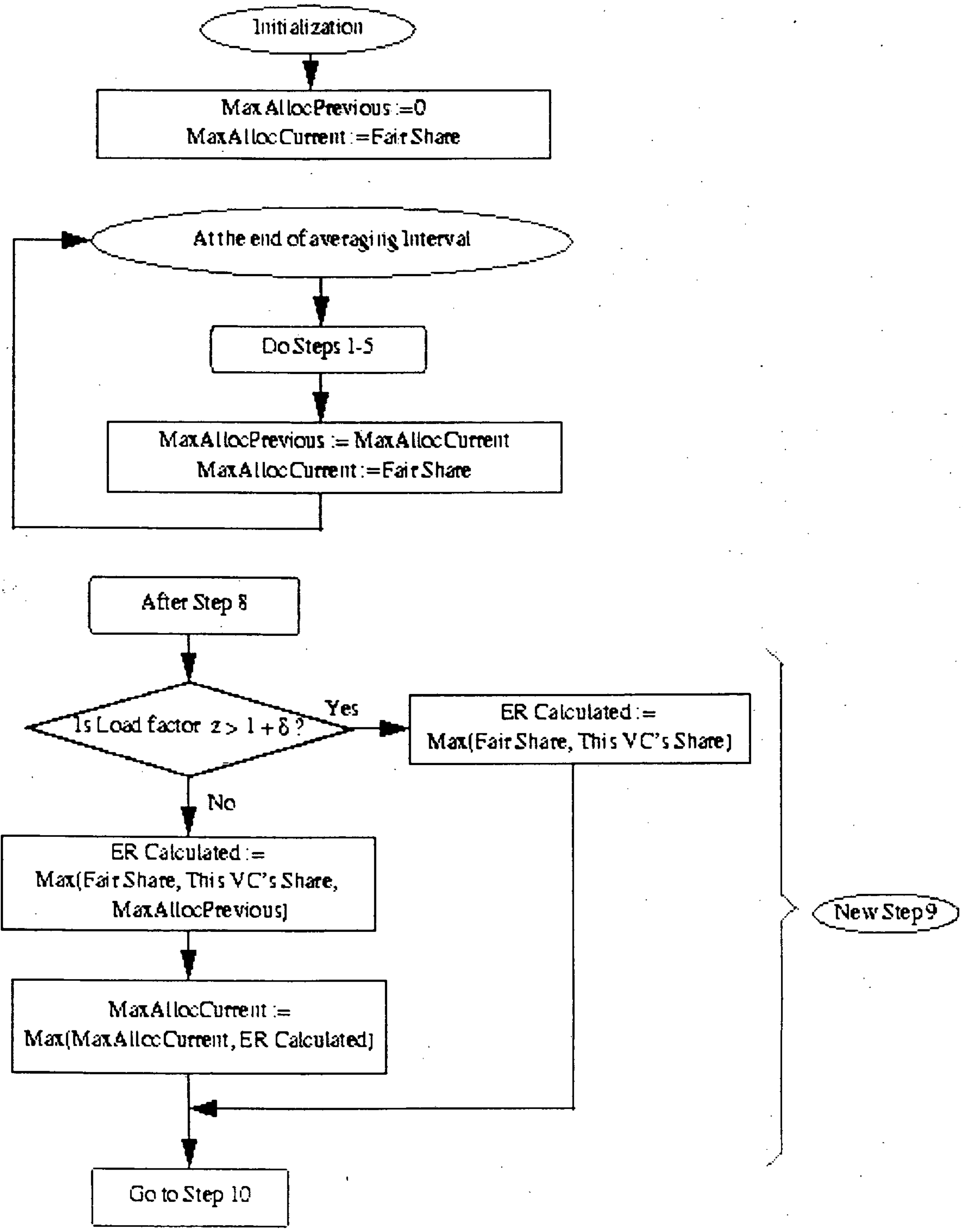
Initialization
MaxAllocPrevious :=0
MaxAllocCurrent :=FairShare
At the end of averaging Interval
Do Steps 1-5
MaxAllocPrevious := MaxAllocCurrent
MaxAllocCurrent :=FairShare
After Step 8
Is Load factor $z > 1 + \delta$?
Yes
ER Calculated :=
Max(FairShare, This VC's Share)
No
ER Calculated :=
Max(FairShare, This VC's Share,
MaxAllocPrevious)
MaxAllocCurrent :=
Max(MaxAllocCurrent, ER Calculated)
Go to Step 10
New Step9

Figure 3 Flow Chart for Achieving Max-Min Fairness

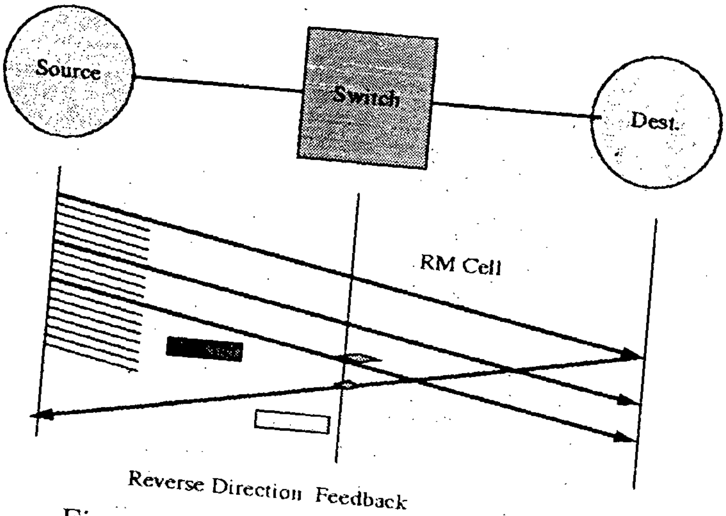

Figure 4: Reverse Direction Feedback

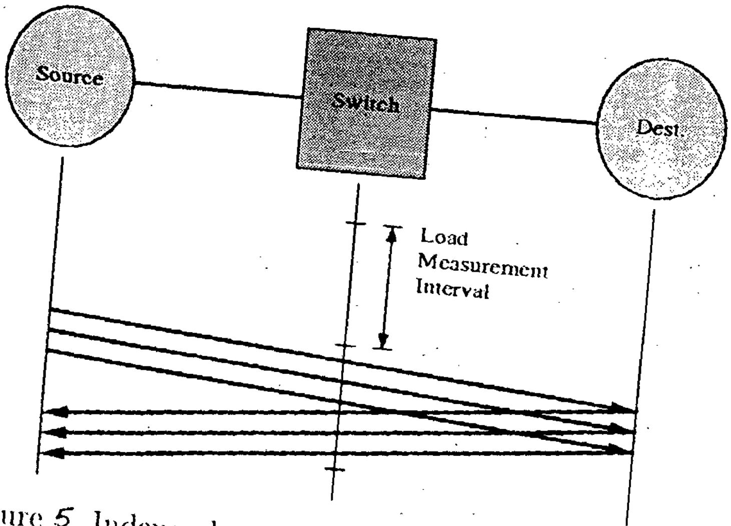

Figure 5 Independence of source and switch intervals

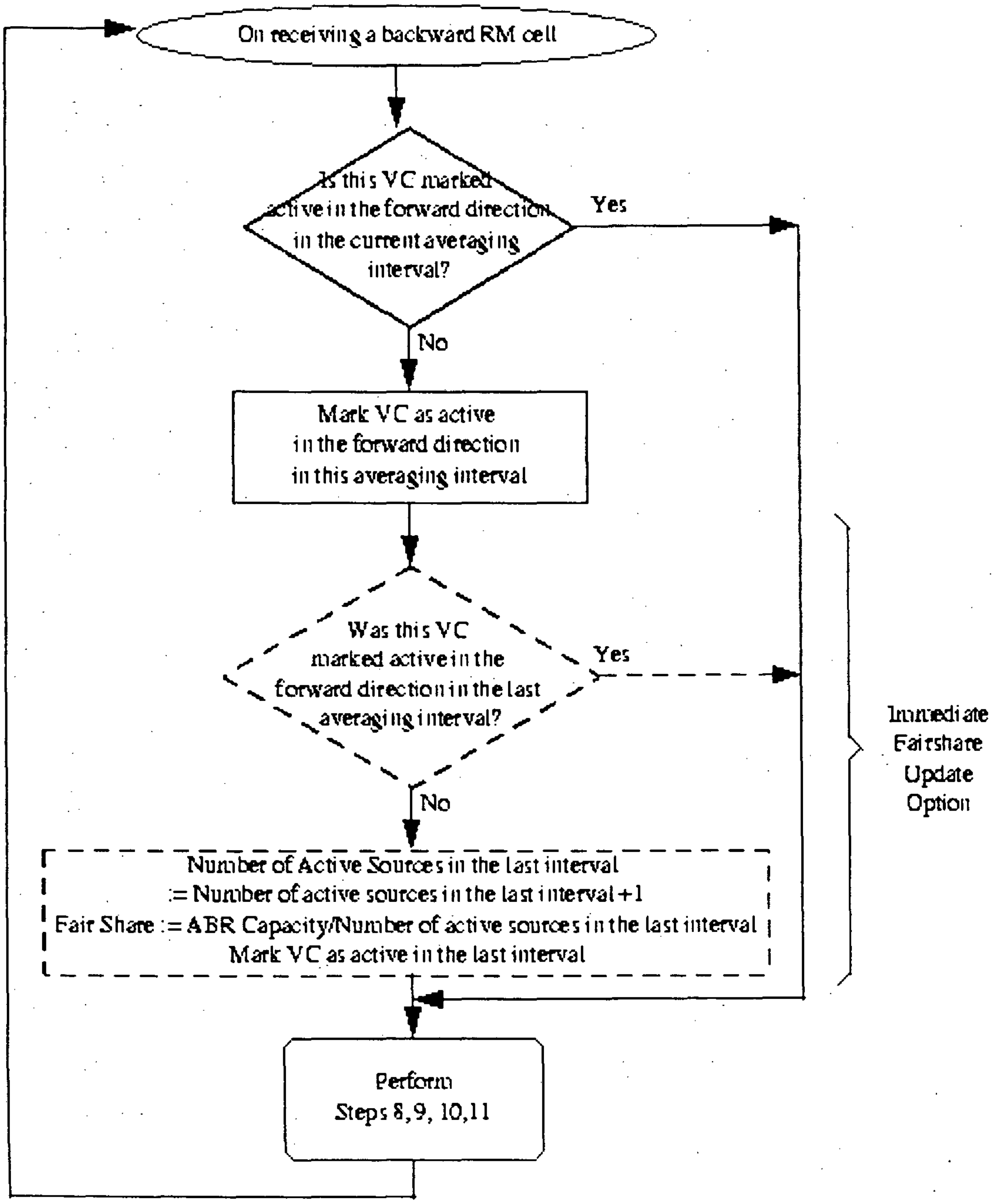
On receiving a backward RM cell
Is this VC marked active in the forward direction in the current averaging interval?
Yes
No
Mark VC as active in the forward direction in this averaging interval
Was this VC marked active in the forward direction in the last averaging interval?
Yes
No
Number of Active Sources in the last interval := Number of active sources in the last interval+1
Fair Share := ABR Capacity/Number of active sources in the last interval
Mark VC as active in the last interval
Immediate Fairshare Update Option
Perform Steps 8,9, 10,11

Figure 6 Flow Chart of Bi-Directional Counting

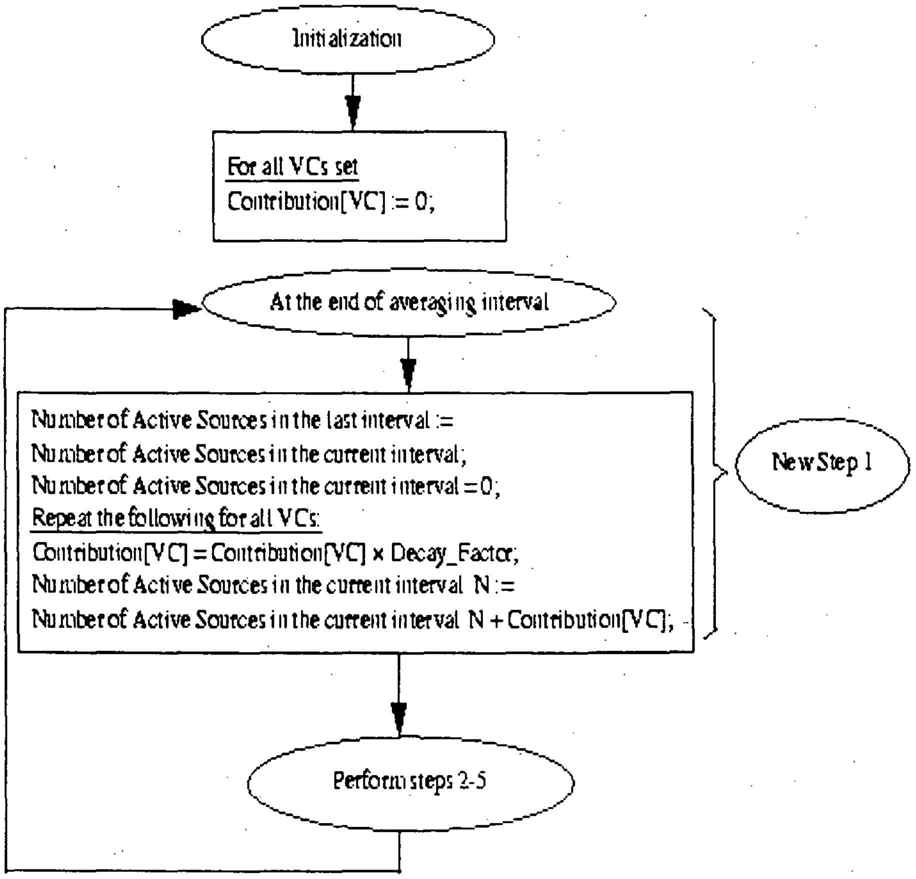
Initialization
For all VCs set
Contribution[VC] := 0;
At the end of averaging interval
Number of Active Sources in the last interval :=
Number of Active Sources in the current interval;
Number of Active Sources in the current interval = 0;
Repeat the following for all VCs:
Contribution[VC] = Contribution[VC] × Decay_Factor;
Number of Active Sources in the current interval N :=
Number of Active Sources in the current interval N + Contribution[VC];
New Step 1
Perform steps 2-5

Figure 7 Flow Chart of averaging number of active sources (part 1 of 2)

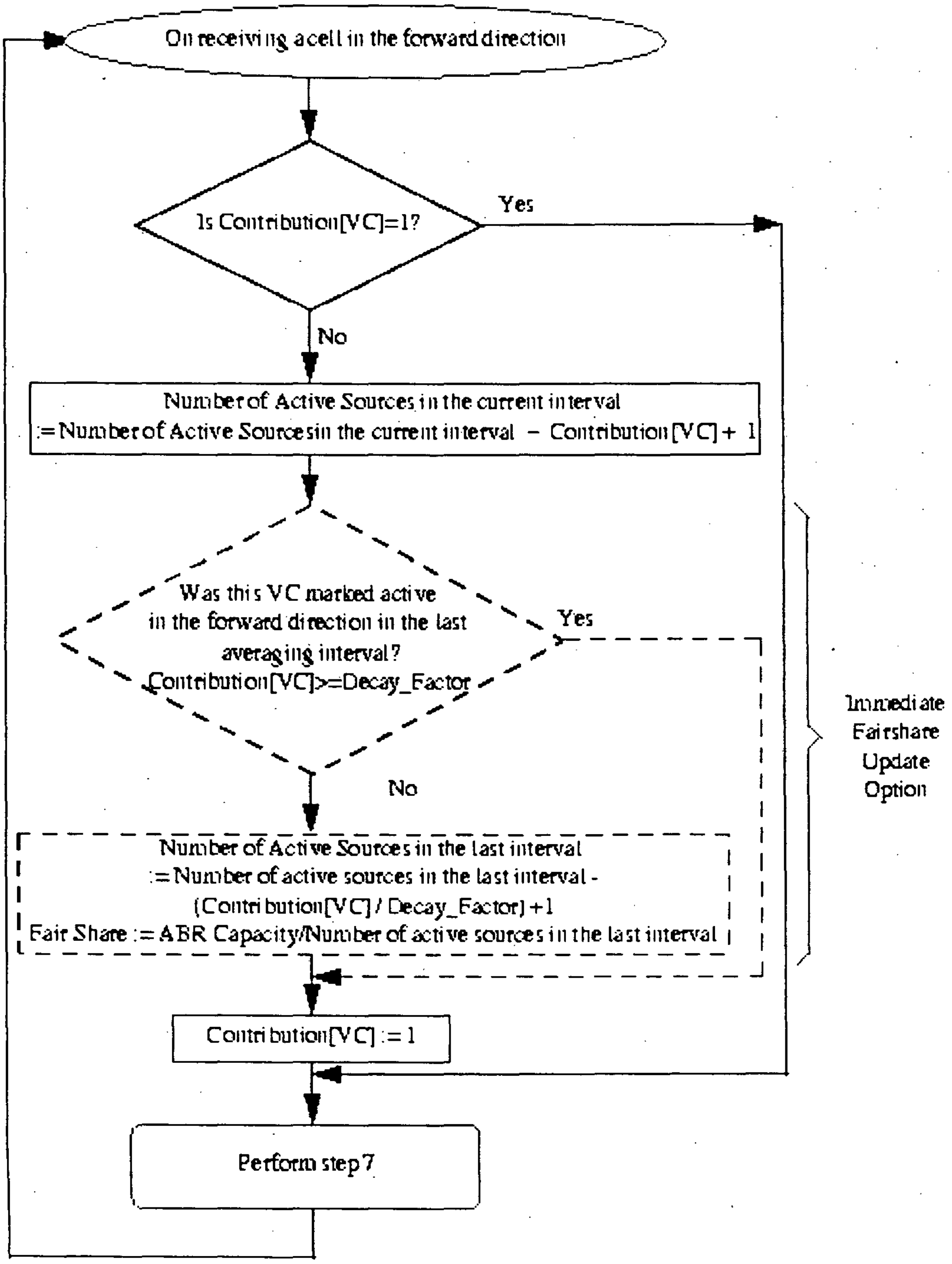
On receiving a cell in the forward direction
Is Contribution[VC]=1?
Yes
No
Number of Active Sources in the current interval
:= Number of Active Sources in the current interval − Contribution[VC] + 1
Was this VC marked active
in the forward direction in the last
averaging interval?
Contribution[VC]>=Decay_Factor
Yes
No
Number of Active Sources in the last interval
:= Number of active sources in the last interval -
(Contribution[VC] / Decay_Factor) +1
Fair Share := ABR Capacity/Number of active sources in the last interval
Immediate
Fairshare
Update
Option
Contribution[VC] := 1
Perform step 7

Figure 8 Flow Chart of averaging number of active sources (part 2 of 2)

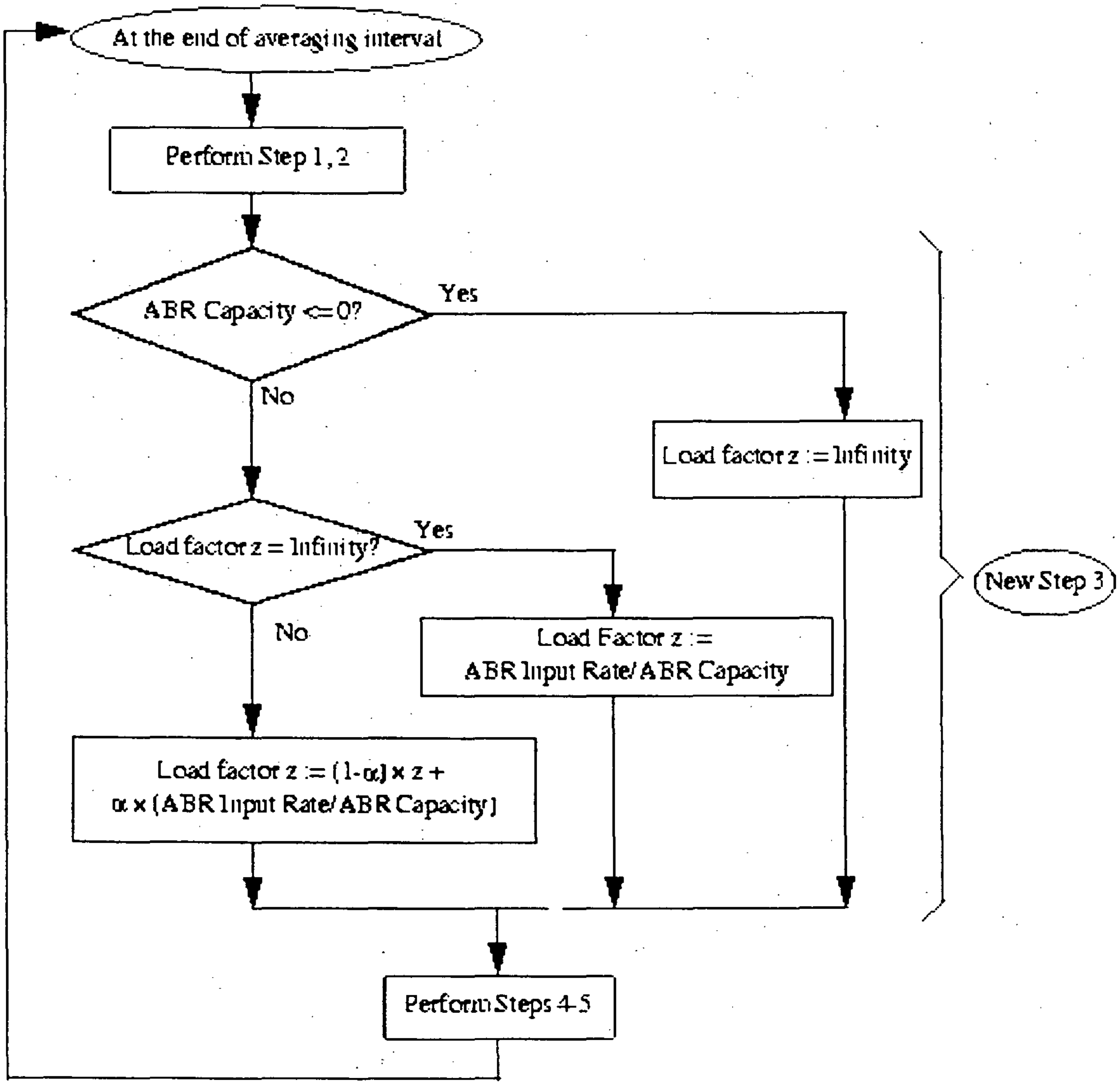
At the end of averaging interval
Perform Step 1,2
ABR Capacity <=0?
Yes
No
Load factor z := Infinity
Load factor z = Infinity?
Yes
No
Load Factor z :=
ABR Input Rate/ABR Capacity
Load factor z := (1-α) x z +
α x (ABR Input Rate/ABR Capacity)
New Step 3
Perform Steps 4-5

Figure 9: Flow chart of averaging of load factor (method 1)

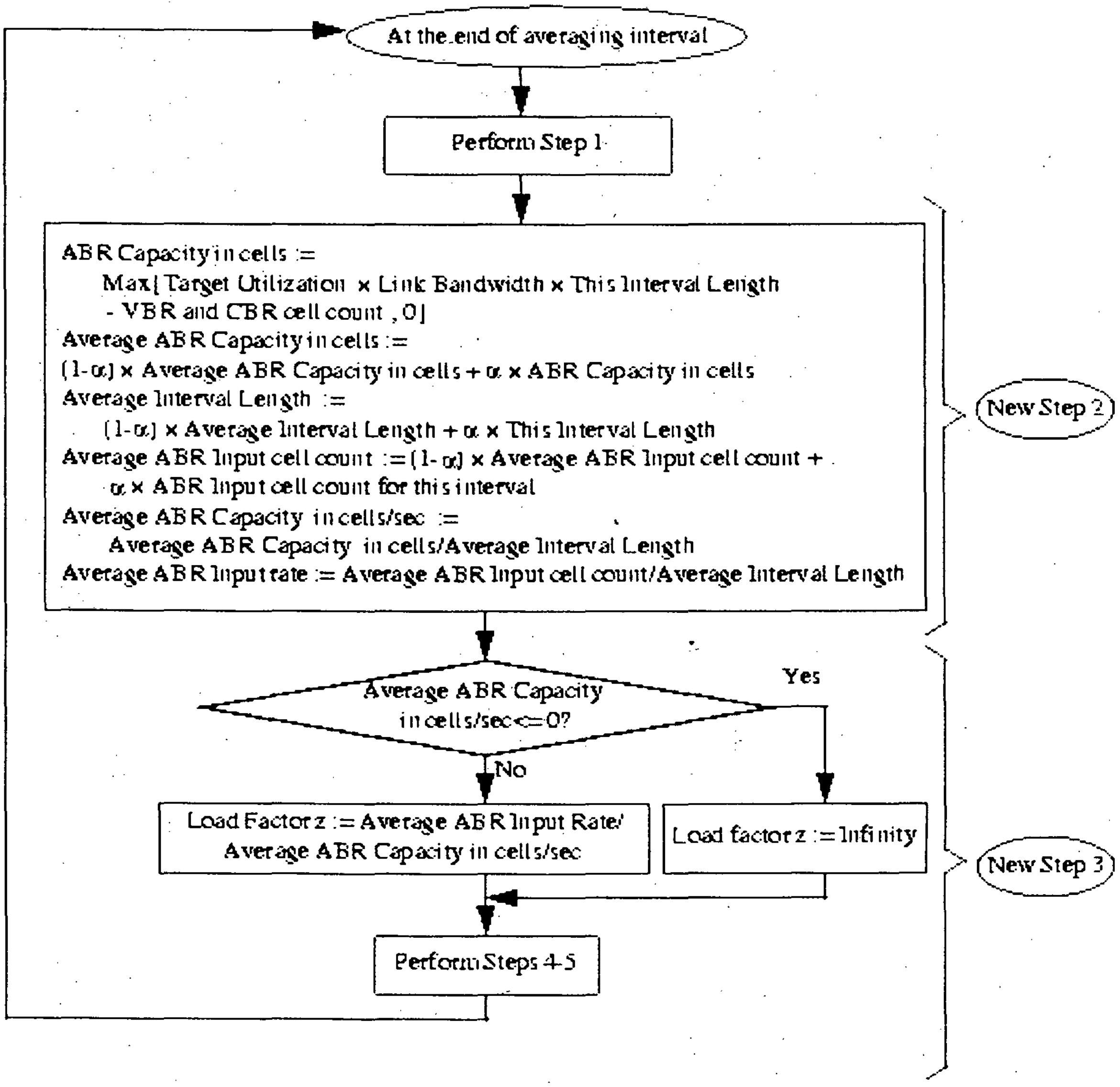
At the end of averaging interval
Perform Step 1
ABR Capacity in cells :=
Max{Target Utilization × Link Bandwidth × This Interval Length
- VBR and CBR cell count , 0}
Average ABR Capacity in cells :=
(1-α) × Average ABR Capacity in cells + α × ABR Capacity in cells
Average Interval Length :=
(1-α) × Average Interval Length + α × This Interval Length
Average ABR Input cell count := (1-α) × Average ABR Input cell count +
α × ABR Input cell count for this interval
Average ABR Capacity in cells/sec :=
Average ABR Capacity in cells/Average Interval Length
Average ABR Input rate := Average ABR Input cell count/Average Interval Length
New Step 2
Average ABR Capacity
in cells/sec<=0?
Yes
No
Load Factor z := Average ABR Input Rate/
Average ABR Capacity in cells/sec
Load factor z := Infinity
New Step 3
Perform Steps 4-5

Figure 10 Flow chart of averaging of load factor (method 2)

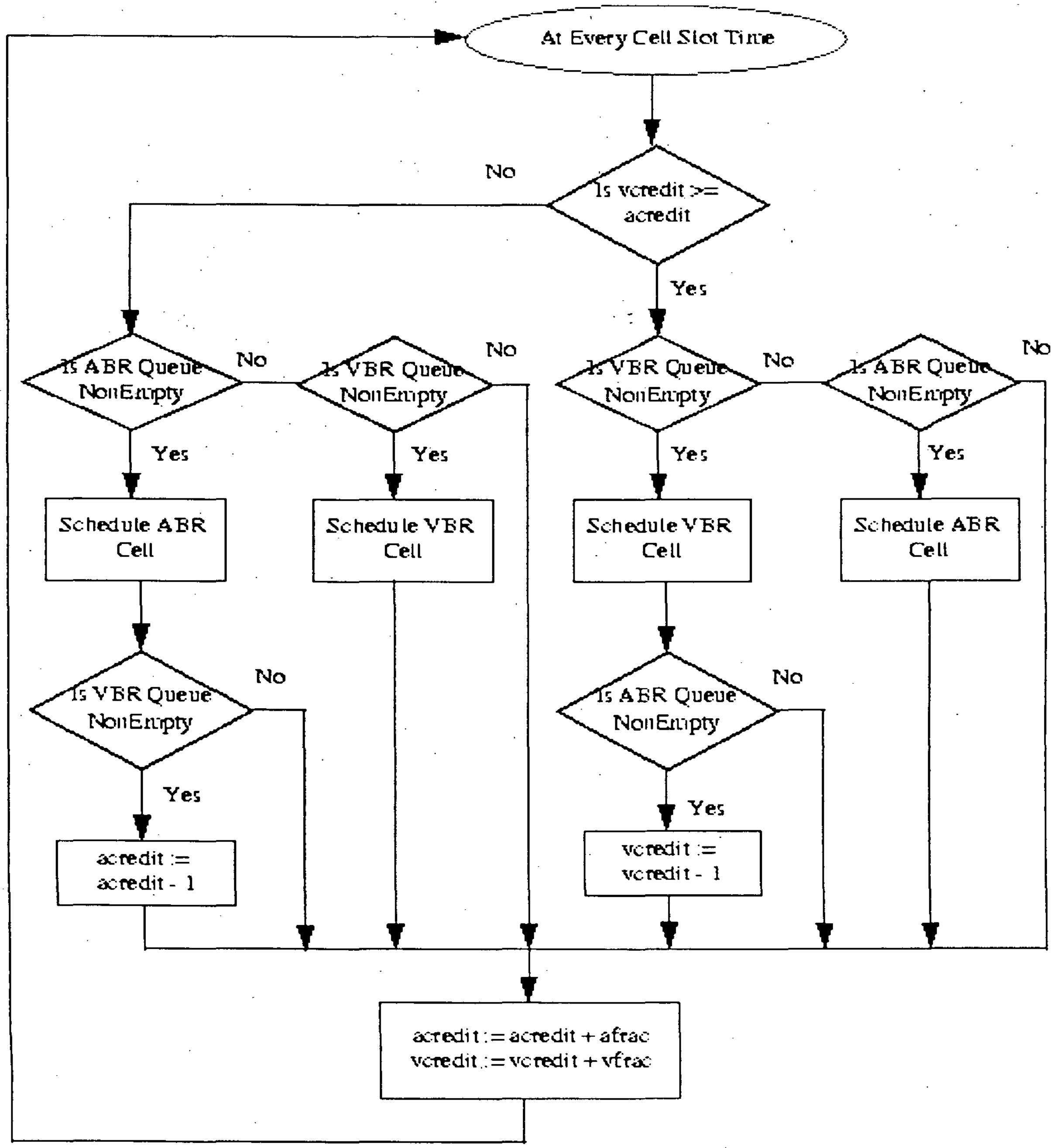

Figure 11 Flow chart of 2-class scheduling