OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

(12) (19) (CA) **Demande-Application**

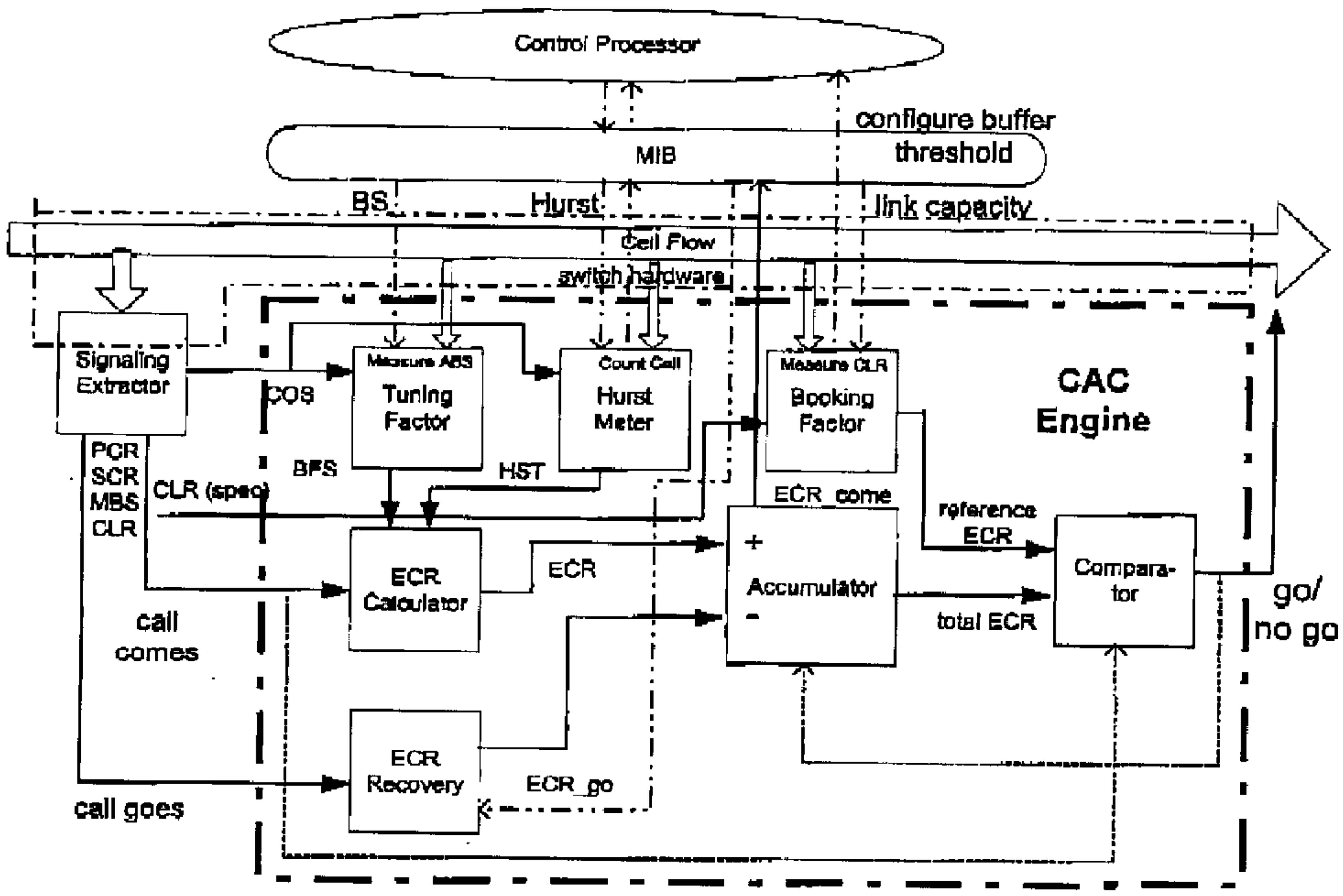
(21) (A1) **2,279,652**
(22) 1999/08/05
(43) 2001/02/05

(72) HUANG, JUN, CA
(72) FENGCHUN, DUAN, CA
(71) SPACEBRIDGE NETWORKS CORPORATION, CA

(51) Int.Cl.[6] H04L 12/12, H04L 12/56

(54) **METHODE ET APPAREIL D'ATTRIBUTION DES RESSOURCES DANS UN RESEAU DE TRANSMISSION DE CELLULES/PAQUETS**

(54) **METHOD AND APPARATUS FOR RESOURCE ALLOCATION IN A CELL/PACKET BASED NETWORK**



(57) A cell/packet based network provided for transmitting Protocol Data Unit (PDU) such as Asynchronous Transfer Mode (ATM) user cells or Internet Protocol (IP) integrated and differentiated service packets. The networks employ a method and apparatus for allocating network resources including uses cell/packet transferring bandwidth in the channel/connection/flow set up phase. The calculation of effective bandwidth is done by hardware using additive-Fractal and measurement-based approach to shorten the processing time and to increase the overall system reliability and efficiency.

ABSTRACT

A cell/packet based network provided for transmitting Protocol Data Unit (PDU) such as Asynchronous Transfer Mode (ATM) user cells or Internet Protocol (IP) integrated and differentiated service packets. The networks employ a method and apparatus for allocating network resources including user cell/packet transferring bandwidth in the channel/connection/flow set up phase. The calculation of effective bandwidth is done by hardware using additive-Fractal and measurement-based approach to shorten the processing time and to increase the overall system reliability and efficiency.

TITLE

Method and Apparatus for Resource Allocation in a Cell/Packet Based Network

FIELD OF INVENTION

The present invention relates to the connection admission control (CAC) of a Broadband Integrated Service Digital Network (B-ISDN) using either cell/packet-based technologies of Asynchronous Transfer Mode (ATM) switch designs for both satellite and terrestrial applications, or new Internet Protocol (IP) router designs with Grade of Service (GoS) concepts such as differentiated service or Multi-Protocol Label Switching (MPLS) architectures.

BACKGROUND OF THE INVENTION

New generations of broadband ISDN technologies that integrate all of the communications services required by the user (i.e., voice, data, moving images, etc.) have recently emerged in communications networks. Traffic information from a large number of calls is transported via fixed-sized packets (or cells in ATM terminology) and then multiplexed onto a single medium whose capacity (in bit per second) is either expressed in a physical size (i.e., link capacity) or a virtual quantity (i.e., effective bandwidth allocation). At connection set-up, the user declares a set of traffic parameters and a required Quality of Service (QoS) from which the carrier must decide whether or not it can establish a path of sufficient bandwidth between the sender and receiver that will satisfactorily accommodate the user's requirement. Connection Admission Control (CAC) is the traffic control function within the network that performs this decision. CAC typically depends on a resource allocation algorithm that determines if there are sufficient network resources (including bandwidth and buffer memory) throughout the communication path. In the event of insufficient resources, the connection is denied. Some prior art CAC methods calculate the Effective Cell Rate (ECR) using complex traffic models. These are usually implemented in software and involve an appreciable computation time. Other CAC methods calculate the cell loss rate of the network using fuzzy logic or other empirical approaches to speed up the computation.

Several problems exist in the complex prior art solutions. These are:

CA 02279652 1999-08-05

1) Speed

Existing CAC methods, such as those described in references [1], [3] and [5] listed below are too complicated for real time applications, e.g. Terabit switch/router designs, or LEO (Low Earth Orbiting) satellite access/switching applications, where a relatively large amount of connections need to be processed/handed-off within a few seconds. In these cases, an extra fast, low-level CAC method suitable for hardware implementation is desirable.

2) Robustness

When the network is down for some reason, the existing CAC methods described in [1], [3] and [5] are not sufficiently robust to recover the same connection on a link because of their complicated non-additive method for calculating the effective cell rate.

3) Flexibility

Existing CAC implementations do not accommodate both Fractal (self-similar) and non-self-similar traffic at the same time. References [1] and [5] on the one hand describe a CAC implementation for non-self-similar cases only while, on the other hand, reference [3] describes a case for a self-similar application. No report, including reference [3], has yet to describe the implementation of self-similar cases. Given that the data and video users are self-similar in nature (as shown in Norros's paper [3]) and that these markets are expanding at a considerable pace, a CAC implementation that writes both self-similar and non-self-similar traffic at the same time is very desirable.

4) Reliability

Existing CAC methods, such as those described in reference [1], [3] and [5], are considered insufficiently reliable for mission critical applications, such as telebanking and telemedicine. For example, a careless or malicious user may provide a set of traffic parameters that could render the complicated algorithms for ECR calculation invalid. This could cause the processor that performs the computation to crash.

5) Accuracy

In accordance with ATM Forum Specifications as described in reference [4], Average Burst Size (ABS) is not included as a traffic descriptor and is not taken into account by other CAC methods, such as those in reference [1], [3] and [5]; however, ABS is considered an important parameter for ECR calculation. Existing CAC methods assume that ABS is the same as Maximum Burst Size (MBS) which obviously it is not. Because of this, existing CACs based on only the input descriptor MBS over-estimate the effective bandwidth, leading to low bandwidth and buffer efficiency.

6) Automation

Existing CAC methods such as those in reference [1] and [5], need manual field tuning by the network operator. Since the operator has to be trained to know how to tune the CAC method this is not considered a cost-effective implementation for network operation.

The following is a description of two prior art CAC methods.

Example A: ICAC

The algorithm is known as IBM CAC.

A complicated EB (equivalent bandwidth) formula assuming Markovian traffic was derived from On-Off Model as:

$$EB = BF \times ((a - BS) + \sqrt{(a - BS)^2 + 4 \times BS \times a \times r}) \times PCR / (2 \times a)$$

Where BS stands for Buffer Size, BF stands for Booking Factor, which can be used to adjust the aggressiveness of the calculation, the default value is to be engineered. Where a is

$$a = \ln(1 / CLR) \times b \times (1 - r) \times PCR$$

And

$$b = MBS / PCR$$
$$r = SCR / (PCR - SCR)$$

In the case of N sources,

$$EB = \min\{\rho + \sigma\sqrt{-2 \times \ln(1 / CLR) - \ln(2\pi)}, \sum_{i=1}^{N} EB_i)$$

Example B: VCAC

The algorithm is still in research community, it is from VTT.

A complicated EB (equivalent bandwidth) formula assuming Fractal traffic can be derived from Fractional Brownian Model as:

$$EB = BF \times (SCR + H(1 - H)^{(1-H)/H} (-2 \ln CLR)^{1/(2H)} (TFa)^{1/(2H)} BS^{-(1-H)/H} SCR^{1/(2H)}$$

Where BS stands for Buffer Size, H is Hurst parameter derived from service and traffic parameter, which could ranges from 0.7 to 0.9 typically. TF stands for Tuning Factor and BF stands for Booking Factor, which can be used to adjust the aggressiveness of the calculation, the default value is to be engineered. In most cases, we will leave it untouched on the field. "a" is the variance coefficient may be derived from the concept of local Hurst as

$$a = (PCR \times ABS)^{H^{H}}$$

Where ABS stands for Average Burst Size.

PRIOR ART REFERENCES

[1] Harry G. Perros, Khaled M. Elsayed, "Call Admission Control Schemes: A Review," IEEE Communications Magazine, November 1996.

[2] Mehrvar, H.R., Le-Ngoc, T., "Estimation of Degree of Self-Similarity for Traffic Control in Broadband Satellite Communications," CCECE'95, Montreal, September 5-8, 1995.

[3] I. Norros, "On the use of Fractional Brownian Motion in the Theory of Connectionless Networks," JSAC, Vol. 13, No. 6, pp. 953-962, 1995.

[4] ATM Forum, UNI3.0 Specification, 1993.

[5] R.J. Gibbens and P.J. Hunt, "Effective Bandwidths for the Multi-type UAS Channels" Queueing Systems 9, pp. 17-28, 1991.

SUMMARY OF THE INVENTION

An object of the present invention is to solve the fore-mentioned six major problems in CAC designs by providing an improved algorithm that satisfies both self-similar and non-self-similar traffic requirements in emerging broadband ISDN networks, such as virtual satellite networks and new internet protocol designs.

The following improvements are made to existing CAC designs:

1) <u>Speed</u>

This invention shows an innovative approach to using logarithmic principles to speed up the CAC computation within a minimum of hardware. The achieved speed is unmatched by traditional software approaches.

2) Robustness

This invention simplifies the ECR calculation using a pure additive formula and makes it relatively easy for hardware implementation. As well, the hardware implementation will provide the recovery of connection in case of a link failure. This feature may not be possible to obtain when using traditional non-additive computation.

3) Flexibility

This invention solves the Fractal (self-similar) implementation problem and works for both self-similar and non-self-similar traffics. Either case is detected automatically by the Hurst value measurement. Moreover, this unified approach offers relatively better speed and a less complicated implementation.

4) Reliability

The number of mathematical operations used in this invention is minimized and greatly simplified. The reliability against careless and malicious users is increased by the simple implementation.

5) Accuracy

The invention takes into account that average burst size (ABS) and maximum burst size (MBS) are different parameters. The hardware automatically detects, measures and calculates the ABS and Hurst value for each applicable service class. The ECR calculation that is based on this measurement will avoid either over-estimation (which will waste system resource) and under-estimation (which will jeopardise the system performance).

6) Automation

This invention uses automated measurements to resolve the problem of inefficient manual operation. Essentially a plug-and-play CAC engine capable of detecting QoS or GoS is used. This is true for newly emerging "Voice over IP" CAC as well, because both the premium and assured flows are characterized by only one or two parameters. Measurement-based methodology works better than parameter-based one. Booking factor, Tuning Factor, and Buffer Threshold are adjusted accordingly.

CAC is a critical function within the Traffic Management (TM) process as described in ATM Forum, UNI3.0 Specification, 1993, and TM is an inseperable feature of ATM technology. The present invention leverages ATM technology to provide and improved overall system. The principle of this invention applies to any packet oriented network that has Quality of Service (QoS) or Grade of Service (GoS) attributes such as integrated IP service, differentiated IP service, or Multi-Protocol Lable Switching (MPLS) service.

BRIEF DESCRIPTION OF THE FIGURES

Figure 1 is an overall block diagram for a CAC system in accordance with this invention;
Figure 2 is a functional diagram for the booking factor (BF) block known in Figure 1;
Figure 3 is a functional diagram for the tuning factor (TF) block known in Figure 1;
Figure 4 is a functional diagram for the Hurst meter block known in Figure 1;
Figure 5 is a functional diagram for the ECR calculator block known in Figure 1;
Figure 6 is a logical diagram for log or anti-log block for use within the CAC system of Figure 1;
Figure 7 illustrates details of the reference ECR calculation within the BF block of Figure 2;
Figure 8 illustrates details of the reference BFS calculating the TF block of Figure 3;
Figure 9 illustrates details of the reference calculating exponent within the ECR calculator block of Figure 5;
Figure 10 illustrates details of the reference calculating base and power within the ECR calculator block of Figure 5;
Figure 11 illustrates details of the reference the final stage of calculating ECR within the ECR calculator block of Figure 5.
Figure 12 is a graph of CLR vs Hurst for 90% load with 100 cell buffer; and
Figure 13 is a graph of CLR vs Load for VBRI with 32 cell buffer.
Table 1 compares the number of operations in this invention relative to prior art.
Table 2 compares the output value in this invention relative to prior art.
Tables 3a-3g compares the variable bit rates (VBR) in this invention relative to prior art.
Table 4 shows buffer size values for different loads in various CBR models.
Table 5 give the cell rate and initial jitter for various CBR models.
Table 6 gives the buffer size values for different loads in each UBR model.
Table 7 shows VBR models for OC3 links.

Table 8 compares simulated CLR with analyzed CLR for different Hurst values.

Tables 9a and 9b compare parameters in different service categories.

DETAILED DESCRIPTION OF THE INVENTION

1. Terminology

Throughout this description the following terms are used in accordance with their respective definitions given below.

Data Unit:	a protocol data unit (PDU), e.g. an ATM user cell or IP user packet
Data Network:	a cell/packet network, e.g. ATM or IP network for serially transmitting data units
Network Element:	ATM switch or IP router
Network Performance:	Quality of Service for each Virtual Channel (VC) or Grade of Service (GoS) for each flow that is to be admitted
Effective Bandwidth:	Effective cell (cells/second) or data rate (bytes/second)
Traffic Characteristics:	The Burstiness or the intensity of traffic fluctuation including ABS and Hurst

2. First Embodiment

In a cell/packet based network including ATM and IP networks for all service categories of transmitting PDU, each ATM switch or IP router has to allocate a certain amount of bandwidth and/or buffer just enough to guarantee that the QoS for each VC (or GoS for each flow) will be satisfied. In a preferred embodiment of this invention, a method is provided for allocating such an effective cell/data rate for buffers as follows:

Connection Admission Control is a key preventative mechanism for cell/packet based network congestion. It is used at connection (or flow) setup in order to determine whether the new connection or flow can be accepted in terms of both its own service requirements and any impact it may have on already existing connections or flows. Logically CAC must be performed for each statistical element in the network. In the satellite environment, these elements include the transmitting queue in the ground terminal, uplink ingress queue, fabric queue, down link queue in on-board switch, and receiving queue in the ground terminal. For each element, a CAC engine can be

CA 02279652 1999-08-05

implemented as described blow. This engine can be cast in a FPGA (Field Programmable Gate Array) or an ASIC (Application Specific Integrated Circuit) chip. Figure 1 is a top-level block diagram of a switching element in which incorporates a CAC engine in accordance with this invention as shown within the dotted block.

Traditionally the CAC physically resides in a real time processor, and logically sits between the user and the network interface and interacts with the software part of the signalling and the Management Information Base (MIB) of the switch. The CAC described under this invention has additional interactions with the switch hardware in order to complete all measurements. The Control Processor in Figure 1 is the real time processor that is responsible to do top level call processing and interactions between all of the elements described above. The MIB is responsible for recording all call processes related to the system and traffic parameters. The switch hardware is the physical board where the cell flow is processed.

The Signalling Extractor block is responsible for extracting traffic parameters such as PCR (Peak Cell Rate), SCR (Sustainable Cell Rate), MBS, QoS (Quality of Service), CLR (Cell Loss Ratio), and the service parameter CoS (Class of Service) from the cell flow, during the call set up phase. It is usually implemented in software in order to increase the number of calls that can be handled. Hardware implementation of this extractor is also possible.

In this embodiment, the CAC engine consists of the following blocks:

a) The Tuning Factor (TF) block is responsible for fetching resource parameters such as Buffer Size (BS) from the Management Information Base (MIB) and fetching and measuring Average Buffer Size (ABS) and Maximum Buffer Size (MBS) from the cell flow. It then calculates the ratio ABS/MBS as the TF and the output ratio BS/TF = BFS (Buffer with Factor Scale Size). The purpose of this calculation is to minimize the operator's manual involvement, and to provide a more automatic solution.

b) The Hurst Meter block is responsible for measuring the Hurst parameter from the measured Index of Dispersion for Count (IDC) of the recorded cell count. The purpose of using the block is to minimize the operator's involvement as well.

c) The Booking Factor (BF) block is responsible for fetching available resource parameters such as Link Capacity from the MIB, and measuring the actual cell loss ratio (CLR_measure). It then increases or decreases the BF by a pre-defined amount, typically

CA 02279652 1999-08-05

around 10%. The lower bound of BF could be in the order of 10%, whereas the upper bound could be in the order of 1000%. If the upper bound of CLR_measure is reached but still does not meet the required CLR specification (CLR_specification), then the BF block increases the Buffer Threshold for the Class of Service (CoS) in use. This will undesirably decrease the Buffer Threshold of the next low class by pre-defined configuration settings, typically 10 cells at a step. This situation is only expected when the traffic profile has major changes. If required the BF maybe over-written by the network operator, but the buffer threshold settings are not to be changed.

d) The Comparison block compares the available link capacity with the total bandwidth required. If the required bandwidth is less than the link capacity, the new call is accepted and the previous total bandwidth is updated; otherwise the call is rejected and the previous total bandwidth is not updated.

e) The Accumulator block is responsible for keeping track of the total required bandwidth. When the call is accepted the ECR and ECR_come indication will be recorded in the MIB for later use and the accumulator is updated. When the call is rejected, both of the above actions are not executed.

f) The ECR Recovery block is a simple but important block that is responsible for retrieving a ECR_go indication from the MIB when the call is to be released in order to update the total bandwidth, or for recovering all connections when there is a link failure.

g) The ECR calculator is a critical block, that takes all of the inputs from both the user and the switch, then produces a single output as Effective-Cell-Rate value, for every call set up request.

3. Improved CAC Method

(a) Markovian Queue Mode

Prior art methods for calculating Equivalent Bandwidth (EB) are typically complicated, computationally difficult, and depend greatly on mathematical assumptions. Yet, in the field, the operator is still required to manually adjust the CAC to meet the real traffic situation as traffic profiles evolve, rendering the sophisticated calculations meaningless. The present invention overcomes this problem and offers a "back-to-basics", very simple, self-tuned hardware CAC implementation which needs little or no manual

CA 02279652 1999-08-05

tuning in the field. A simple equivalent bandwidth formula, expressed in Effective Cell Rate (ERC), is obtained from a Markovian queue model as:

(a) $ECR = BF x SCR \div (CLR x SCR \div PCR)^{(MBS x TF \div BS)}$

where BS is the Buffer Size for a theoretical queue, MBS is the Maximum Burst Size, TF is the Tuning Factor and BF is the Booking Factor. The latter two factors are used to adjust the aggressiveness of the calculation. The switch vendor provides the initial default values, while the real values are tuned automatically in the field. In most cases, these values are left untouched in the field unless the network operator wants to override the settings. It should be noted that, since the CAC has to be exercised for each service class of queues, in actual practice the theoretical buffer size (BS) corresponds to the buffer threshold for each service class.

A multiplexing of large amount of connections is modeled as a fundamental M/M/1 queue, where we have:

$$CLR \approx (SCR \div EB)^{(BS \div ABS)}$$

Here SCR/EB is system load. BS/ABS is the equivalent number of seats in a queuing system. When the number of connection is not very large, the aggregated traffic may not be as smooth as in Poisson distribution. We therefore have to take the burstiness PCR/SCR into consideration. Apparently cell loss is proportional to the burstiness:

$$CLR \cong PCR \div SCR \times (SCR \div EB)^{(BS \div ABS)}$$

Taking EB as the unknown, we solve the equation and get:

$$EB \cong SCR \div (CLR \times SCR \div PCR)^{(ABS \div BS)}$$

Because the theoretical assumption may not capture all the delicate aspects of real traffic and network (such as long term dependence of traffic, the impact of constant change of life style on the traffic profile, unforeseeable network signaling, control, and maintenance traffic) we still need a tunable Booking Factor to adjust the final calculation on the field. The default value for BF can be 1.

Since ABS is not a traffic descriptor, MBS (Maximum Burst Size) is. A simple way to map is ABS=20%MBS. The number can be tuned in the field as well.

This is where the final simple form of the EB (or ECR) Markovian formula (equation a) comes from.

(b) Self-similar Model

Self-similar traffic characteristics are getting more and more attention from the major carriers as the Internet market is growing. A simple equivalent bandwidth formula assuming a Fractal traffic, is further derived from a Fractional Brownian Motion (FBM) model based on the previous formula using wavelet like method as:

(b) $ECR=SCR \div (CLR \times SCR \div PCR)^{(MBS \div (BS \div TF)) \div (2-2H)}$

(c) $Refrence\_ECR=Link\_Cap/BF$

Where H is the Hurst parameter obtained from statistical measurement for applicable traffic with certain service and traffic parameters. These could range from 0.5 to 0.99, with typical value of 0.7 to 0.9 for self similar traffic and 0.5 for non-self-similar traffic.

It is not necessary to do a mathematical operation on BF for every call set up since the updating of BF is done on a periodical (hourly or daily) basis. BF operations can be equivalently obtained on the link capacity instead of the traffic bandwidth (ERC) itself. It should also be noted that, for future MPLS applications, the above formula is applicable as well, except that the unit would be in bytes/second instead of cells/second, and the maximum burst size would correspond to the maximum packet size.

4. Detailed Derivations

(a) Self-similar Algorithm

A simple equivalent bandwidth (EB) formula assuming Fractal traffic is derived from Fractional Brownian Motion (FBM) Model as:

$$EB = BF \times SCR \div (CLR \times SCR \div (TF1 \times PCR))^{(TF2 \times MBS \div BS)^{1-TF1 \times (2H-1)}}$$

Where H is Hurst parameter derived from service and traffic parameter, which could ranges from 0.5 to 0.99 with typical value of 0.7 to 0.9 for self similar traffic and 0.5 for non-self-similar traffic. The Booking Factor (BF) is used to adjust the aggressiveness of the calculation. The default value is normally engineered and in most cases, it is left untouched in the field unless the network operator wants to change it.

A multiplexing of a large amount of connections is modeled as a fundamental FBM/M/1 queue, as follows:

$$CLR \approx (e^{-c(\rho)})^{(BS \div ABS)^{2-2H}}$$

where BS/ABS is the equivalent number of seats in the queuing system, and

$$c(\rho) = n^{2H-1}((1-\rho)/\rho)^{2H}(((1-H)/H)^{H} + ((H/(1-H))^{(1-H)})^{2}/2a$$

With some approximation we have

$$e^{-c(\rho)} \approx \rho = SCR / EB$$

where SCR/EB is the system load. Taking the burstiness PCR/SCR into consideration, we get:

$$CLR \cong PCR \div SCR \times (SCR \div EB)^{(BS \div ABS)^{2-2H}}$$

Solving for the unknown EB, we get:

$$EB \cong SCR \div (CLR \times SCR \div PCR)^{(ABS \div BS)^{2-2H}}$$

This equation is the final enhanced simple form of the EB formula presented at the beginning of this section.

Comparing with the basic algorithm, only one more operation been added. Follwing is the pseudo code for the actual implementation:

(b) Pseudo code

/* Pseudo code for effective cell rate calculations */

//sanitary checking

If any of inputs=0,

Report error.

//some pre-calculations needed only when network system is reconfigured

BF=1; //BF-Booking Factor

CLS=CLR/TF1; //CLS-Cell Loss Scaled; CLR-Cell Loss Ratio; TF1-Tuning Factor

BFS=BS/TF2; //BFS-Buffer Size Scaled; TF2-Tuning Factor

HST=1-TF3*(2H-1); //HST-Hurst Scaled & Tuned; H-Hurst; TF3-Tuning Factor

Scale=128/3864; //For Flight Mode, interfacing ground segment

// Loop for every call set up

$ECR = BF \times SCR \div (CLS \times SCR \div PCR)^{(MBS \div BFS)^{HST}}$ //ECR-Effective Cell Rate;

//SCR-Sustainable Cell Rate;

//PCR-Peak Cell Rate;

//MBS-Maximum Burst Size;

// Convert cells/s to slot/frame (may skip this for engineering model)

ECR=ECR/Scale;

```
// Adding in new call's contribution
Sum of ECR=Sum of ECR+ECR;
//Decision making
If Sum of ECR< Link speed
Accept call
//Update the record for call release use
ECR(VC)=ECR;
Else
Reject
// Take off new call's contribution
Sum of ECR=Sum of ECR-ECR;
```

The Pseudo code for pre-preprocessing of 8 classes is not included here.

The actual number of Scales is determined by the frame formats; for a MF-TDMA frame with 2Mbps carriers, it is 128/3864. (More details are available in to prior art document: Robert Craig, Radu Irogulescu, "SB2200P Terminal Sub-system Design Document" 02SSD0001, 1998. The other values can be calculated similarly.

5. Alternative CAC Embodiments

CBR (Continuous Bit Rate) CAC

Although PCR maybe used as ECR for CBR connection, CDV is not necessarily guaranteed. Therefore, a safe approach is to calculate the ECR with the CAC method described for the first embodiment. Since CBR is much less bursty than VBR, the following TF set should be used:

TF1=0.1, TF2=1, TF3=1.

And the default traffic parameters are

SCR=PCR, MBS=1, H=0.5.

RT-VBR (Variable Bit Rate) CAC

ECR is calculated by the CAC method described for the first embodiment. Since VBR is much bursty than VBR, the following TF set are used:

TF1=0.2, TF2=0.5, TF3=0.5.

And the default traffic parameters are
H=0.7.

NRT-VBR CAC

ECR is calculated by the CAC method described for the first embodiment. Since VBR is much bursty than VBR, the following TF set should be used:
TF1=0.25, TF2=0.2, TF3=0.7.
And the default traffic parameters are
H=0.8.

GFR/ABR CAC

ECR is calculated by the CAC method described for the first embodiment.. Since GFR/ABR has MFR/MCR guarantee requirement, the following TF set should be used:
TF1=0.3, TF2=1, TF3=1.
And the default traffic parameters are
PCR=SCR=MCR, H=0.5.

UBR CAC

No ECR calculation is needed. For load balancing purpose, we assign ECR=10 cells/second.

A procedure for fine tuning the above recommended TF factors and Hurst parameter is provided in the simulation section further below.

6. Analysis

Table 1 compares the computation requirements among three different CAC algorithms: the present invention – Space Bridge CAC (SCAS), the Markovian CAC (MCAC) and the Fractal CAC (FCAC). It can be seen that our present invention, the Space Bridge/Self-similar CAC embodiment, has considerably less mathematical operations than the other two.

Table 2 compares the computation results (with standard Bellcore's traffic test-set) among the different methods. It should be noted that the ECR values obtained by different methods are quite close, which means that the SCAC method of the present invention agrees with the others. Since different equipment vendors may use different methods, the closer the results, the better the interoperability between switches from different vendors.

Table 3a computes results of VBRI for CAC method of this invention.
Table 3b computes results of VBRI for prior art IBM CAC method.
Table 3c computes result of VBRII for prior art CAC method of this invention.
Table 3d computes results of VBRII for prior art IBM CAC method.
Table 3e computes result of VBRIII for the CAC method of this invention.
Table 3f computes results of VBRIII for prior art IBM CAC method.
Table 3g computes results of VBRIII for prior art VTT CAC method.

For CBR, the prior art formula used is:

$$CLR = -\frac{(1-load)}{\ln(load)}e^{-BS(2BS/N+1-load-\ln(load))}$$

Where N is total number of source.
Assuming CLR is 10^{-10} and 155Mbps link, by using MATLAB, we find the buffer size for different load, or ECR/PCR ratio as shown in Table 4.
Prior art Bellcore CBR models are shown in Table 5 as given in Bellcore, "Broadband Switching System Generic Requirements", BSS Performance part, GR-110-CORE, Sep. 1994.
For VBR, the prior art formula used is the following as found in S. Ganti, "A Comparison of Virtual Bandwidth Algorithms", NNC 31 PTM0051, 1996:

$$ABS = \frac{1}{1-10^{-\eta/MBS}}$$

$$\mu = PCR / ABS$$

$$\lambda = \mu \times SCR / (PCR - SCR)$$

$$\zeta = \ln(CLR) / BS$$

$$ECR = \frac{\zeta \times PCR + \mu - \lambda - \sqrt{(\zeta \times PCR + \mu + \lambda)^2 + 4\lambda\mu}}{2\zeta}$$

Assuming CLR is 10^{-9} and 155Mbps link, by using MATLAB, we find the buffer size for different load, or ECR/PCR ratio as given in Table 6.

Prior art Bellcore VBR models for OC3 links are given in Table 7.

7. Hardware Embodiment

This section describes the function and logic levels of each of the blocks shown in the CAC engine in Figure 1, from Figure 2 to Figure 5, and then describes the basic idea of using logarithmic method as shown in Figure 6. The description applies to a hardware embodiment of this invention. It also applies to other embodiments combining hardware with software.

The BF block, shown in Figure 2, consists of two sub-blocks. The one with a regular rectangle is the foreground processing of the reference of ECR and the one with a curved rectangle is the background processing including statistical measurements. A set of counters is to be maintained for each service class in the switch to record the number of cells being discarded within the switch and the number of cells entered into the switch during a certain period of time. The CLR_measure is the ratio of the number of discarded cells over the number of entered cells. The measurement period can be set to be 1 hour or 1 day or more if the link speed is low.

The TF block is shown in Figure 3. In addition to the tuning of the Booking Factor for bandwidth, the tuning of the Tuning Factor for buffer is also included. Also this operation is done for every call in the TF block, because each class of calls has different traffic characteristics. Similar to Figure 2, the curved rectangle in Figure 3 is for background statistical measurement. A set of counters is maintained for each applicable service class in the switch to record the number of cells entered into the switch and the number of packet entered into the switch during a certain period of time. The ABS is the ratio of the number of cells over the number of packets. The measurement period can be set to be 1 minute or 1 hour. The buffer size, which is actually the threshold can be software configured if necessary.

The Hurst Meter block is configured as shown in Figure 4. A set of counters is maintained for each applicable service class in the switch to record the number of cells entered into the switch during a certain period of time. The numbers are stored in the MIB where the software calculates the H value. The operation of the intermediate parameter HST=2-2H, which represents smoothness of traffic, can be done by either software or hardware. In this embodiment hardware is chosen to ease the software task. H is the slope of the loglog IDC plot plus 1 divided by 2. A common measure for capturing the variability of traffic over different time scales is provided by the Index of Dispersion for Counts (IDC). This formula is denoted as the variance of the number of arrivals during a given time interval of length m divided by the expected value of that same quantity:

$$IDC(m) = \frac{\mathrm{var}\left(\sum_{j=1}^{m} X_j\right)}{E\left[\sum_{j=1}^{m} x_j\right]}$$

Where X_j (j=1,2,....m) is an observed time series and $E[X_j]=\mu$

It should be noted that the cell counter can be shared amongst the TF, BF and Hurst blocks. Therefore, only three sets of counters are needed: cell discard counter, total cell counter and packet counter. But the measurement period has to be stated (e.g. 1 hours). If the switch has various link speeds, separate additional counters are needed.

The ECR calculator is shown in Figure 5, where the Buffer with Factor-Scaled Size (BFS) equals BS/TF. The divider, multiplier and power can be implemented by using commercial of the shelf cell elements in FPGA, such as the one provided by Altera Corporation. However, the present invention further minimizes the computation time and hardware by employing a logarithmic approach.

In this embodiment, logarithms are used to convert the multiplication, division, and power calculation into addition and subtraction calculations in order to simplify the hardware implementation and increase speed. For parameter values ranging from 10^{-10} to 10^{7} it is easier to convert their values into logarithmic values. This will convert the time-consuming multiplications and divisions into faster add and subtraction calculations. Figure 6 shows a pair of logarithmic blocks needed for this operation.

The granularity will determine the size of the table. The size can be designed just big enough to cover the actual operation range with desired accuracy. Typically the size of 1024 will provide 3 effective digits of accuracy. The relative error is expected to be within 1%, which is enough for practical usage. Two tables are used, one is a log table, which is 14 bits per address and a 900(1K) address, and the other table is anti-log table, which is 10 bits per address, and 900(1K) address.

CA 02279652 1999-08-05

8. Design Calculation within Major Blocks in the CAC Engine

Figures 7 and 8 refer to the BF and TF block respectively and Figures 9 to 11 refer to the ECR calculator block.

Figure 7 shows the detail design for the bandwidth reference for ECR using logarithmic calculations for:

Reference_ECR=Link_Cap/BF

A *Reference_ECR* value is calculated in this block. The values of the link-capacity and the Booking Factor, which should be extracted from the cell flow, are converted to the log values by using the log look-up table, respectively. The *Reference_ECR* value is obtained by using the difference of these two values as an address to the anti-log look-up table. The input values of the block are Link-Capacity and Booking Factor. The output of the block is the *Reference_ECR* value.

Figure 8 shows the detailed design for buffer size reference (BFS) using logarithmic calculations for:

BFS=BS/TF=BS/(ABS/MBS)

This block computes a log (BFS) value. The values of Buffer Size (BS), Average Burst Size (ABS), and Maximum Burst Size (MBS) are used to index the log look-up table to obtain log values. The log (BFS) can be obtained by two subtracting calculations. The inputs are BS, ABS and MBS. The output of this block is a log value of BFS.

The following breakdown structure of ECR calculation for each connection is then implemented:

$ECR = SCR \div Power$

$Power = Base^{Exponent}$

$Base = CLR x SCR \div PCR$

$Exponent = MBS \div (BS \div TF) \div (2\text{-}2H)$

Figure 9 shows the implementation of

Log (Exponent)=Log((MBS÷(BS÷Tf) ÷(2-2xH)).

This block consists of a shift register, the three subtractors. In order to get a log(2-2H), a HST is shifted and subtracted by 2, then the result is used as an address to the log table. A log(Exponent) can be obtained by two times the subtract calculation.

Figure 10 shows the implementation of the Power calculations including Base part. The inputs of this block are log(CLR), log(SCR), log(PCR), and log(Exponent), which is

CA 02279652 1999-08-05

the output of Figure 9. The output of this block is a log(Power) value. This block consists of two adders and one subtractor.

Figure 11 shows an implementation of the final stage of the ECR calculation. This block is used to obtain the ECR. A log(SCR) and a log(Power), which is computed by the power calculation block, are the inputs of this block. The result of log(SCR) subtracting log(Power) is used as an address to the anti-log table to fetch the related ECR value.

By counting the longest path in the hardware design, which is from MBS to ECR (Figures 8-11), we find a maximum of 12 clocks are needed to complete the whole processing, with a regular 100 MHz clock rate, such as Alter's FPGA board. This means that only 120 ns is consumed per call. However, even with a 40 MHz real-time software scheme (such as using WindRiver's VxWorks) it was observed that only 15 ticks perform 1000000 mathematical operations, where each tick consumes 1/60 second. Thus 8 mathematical operations will need 2 μs to finish only the computation part. This does not include the comparison and memory ready/writing times, which consume more than the computation times. The assumption is that the whole processor is doing only the computation. We can therefore conclude that the hardware implementation will be one or two magnitudes faster than the software implementation. Assuming 3 digits accuracy, with regular FPGA technology used today, it can be calculated (based on the Altera 10K FPGA specification) that 2400 gates are required to implement tables and 1600 gates to implement the logic. Thus, the total gate counts is only about 4K.

9. SIMULATIONS

9.1 Procedure for Theoretical Simulation

The general method is to simulate a single first-in-first-out (FIFO) queue with a homogenous given input, find the loss, and compare with the analytic one dictated by:

$CLR = load^{BS*2J}$,

where J=(2-2H)/(3-2H). For load=0.9, BS=100, the simulation was obtained from prior art as shown in Table 8 and graphically shown in Figure 12.

9.2 Procedure for Benchmark Simulation

The general method is to simulate a single FIFO queue with homogenous given input, find the loss curve, and matches the analytic one with that of simulated one. The loss for analytic one is given by following formula.

$$CLR = \frac{TF1 \times PCR}{SCR} \times \left(\frac{BF \times SCR}{EB}\right)^{\left(\frac{TF2 \times MBS}{BS}\right)^{TF3 \times (2H-1)-1}}$$

For the simulation, 155Mbps link, T1 CBR, and DS-3 VBR are assumed, the number of source should vary so that the load varies from 0.6, 0.7, 0.8 until 0.9.

CBR CAC

Bellcore CBR model is used to tune following factors, starting with center valus of TF1=0.1, TF2=1, TF3=1.

RT-VBR CAC

Bellcore VBR model is used to tune following factors, starting with center values of TF1=0.2, TF2=0.5, TF3=0.5.
Furthermore, the default traffic parameter is tuned based on FBM model, by using certain average of existing measurement, starting with center value of H=0.7.

NRT-VBR CAC

Bellcore VBR model is used to tune following factors, starting with center values of TF1=0.25, TF2=0.2, TF3=0.7.
Furthermore, the default traffic parameters is tuned based on WWW model, by using certain average of existing measurement, starting with center value of H=0.8.

GFR/ABR CAC

Poisson model or TCP model should be used to tune following factors, starting with scenter value of TF1=0.3, TF2=1, TF3=1.

CA 02279652 1999-08-05

UBR CAC

Poisson or PMGP (Pareto Modulated General Process) model should be used to tune following default parameter, staring with a center value of ECR=10 cells/second.

9.3 Real Simulation

The general method is to simulate a switch with heterogeneous given input, find the loss, and fine-tune the factors. One of results is graphically shown in Figure 13.

10. CAC Embodiments Tailored for Virtual Satelllite Network (VSN)

Random Early Call Rejection (RECR) policy will be used in case there is sharing between VSN. Call will be rejected with certain probability that varies with the total system load, as described in AT&T Lab, IEEE/ACM Networking 1999.

11. CAC Embodiment Tailored for Low-earth-orbiting (LEO) Satellites

Hardware based CAC is preferred. At first call set up time, the ECR is recorded somewhere in a table for hand over process.

12. OTHER INFORMATION

Other information may need to be stored in MIB. For our switch, the network system supports 5 service categories, with 8 distinct priorities, as given in Table 9a.

Link speed, CLR and buffer size need to be fetched from MIB individually as well, as shown in Table 9b.

CA 02279652 1999-08-05

ACRONYMS:

ATM	Asynchronous Transfer Mode
ABS	Average Burst Size
ASIC	Application Specific Integrated Circuit
BF	Booking Factor
BFS	Buffer with Factor scaled Size
BS	Buffer Size
CAC	Connection Admission Control
CLR	Cell Loss Ratio
CoS	Class of Service
ECR	Effective Cell Rate
FPGA	Field Programmable Gate Array
GoS	Grade of Service
HST	Hurst Smoothness of Traffic
IDC	Index of Dispersion for Count
IP	Internet Protocol
MBS	Maximum Burst Size
MIB	Management Information Base
MPLS	Multi-Protocol Label Switching
PDU	Protocol Data Unit
PCR	Peak Cell Rate
QoS	Quality of Service
SCAC	Self-similar CAC
SCR	Sustainable Cell Rate
TM	Traffic Management
TF	Tuning Factor
UNI	User Network Interface
VC	Virtual Channel

CA 02279652 1999-08-05

WHAT IS CLAIMED IS

1. A method of controlling connection admission within a data network for data units each having a service class, comprising the steps of:

a) measuring network performance periodically for each service class and adjusting a booking factor and buffer threshold accordingly;

b) measuring traffic characteristics periodically, for each service class, and adjusting an associated tuning factor accordingly;

c) during connection set up phase, extracting service parameters and traffic parameters, fetching associated network system parameters from a management information base (MIB);

d) calculating an effective bandwidth for transferring the data units within the data network using a self-similar aware additive formula;

e) making a final decision to admit or reject a connection set-up request.

2. A method as in claim 1, further comprising the steps of:

f) updating information for the data network according to said final decision;

g) during a connection release phase or a link failure case, fetching associated effective bandwidth value from the MIB, and updating status information for the data network accordingly.

3. A method as in claim 1, wherein the additive formula is implemented with logarithmic-table based hardware system.

4. A system for connection admission control within a data network for data units each having a service class, said system comprising:

a) means for measuring the network performance periodically for each service class and for adjusting a booking factor and buffer threshold accordingly;

b) means for measuring traffic characteristics periodically, for each respective service class, and for adjusting an associated tuning factor accordinly;

c) means for extracting service parameters and traffic parameters and for fetching associated network system parameters from a management information base (MIB) during connection set up phase;

d) means for calculating an effective bandwidth for transferring the data units within the data network, using a self-similar aware additive formula;

e) a decision block for admitting or rejecting a connection/set-up request.

5. A system as in claim 4, further comprising:

f) means for updating the information for the data network according to said final decision;

g) means for fetching associated effective bandwidth value from the MIB and for updating the information for the data network accordingly, during a connection release phase or a link failure case.

6. A system as in claim 4, wherein the additive formula is implemented with logarithmic-table based hardware.

CA 02279652 1999-08-05

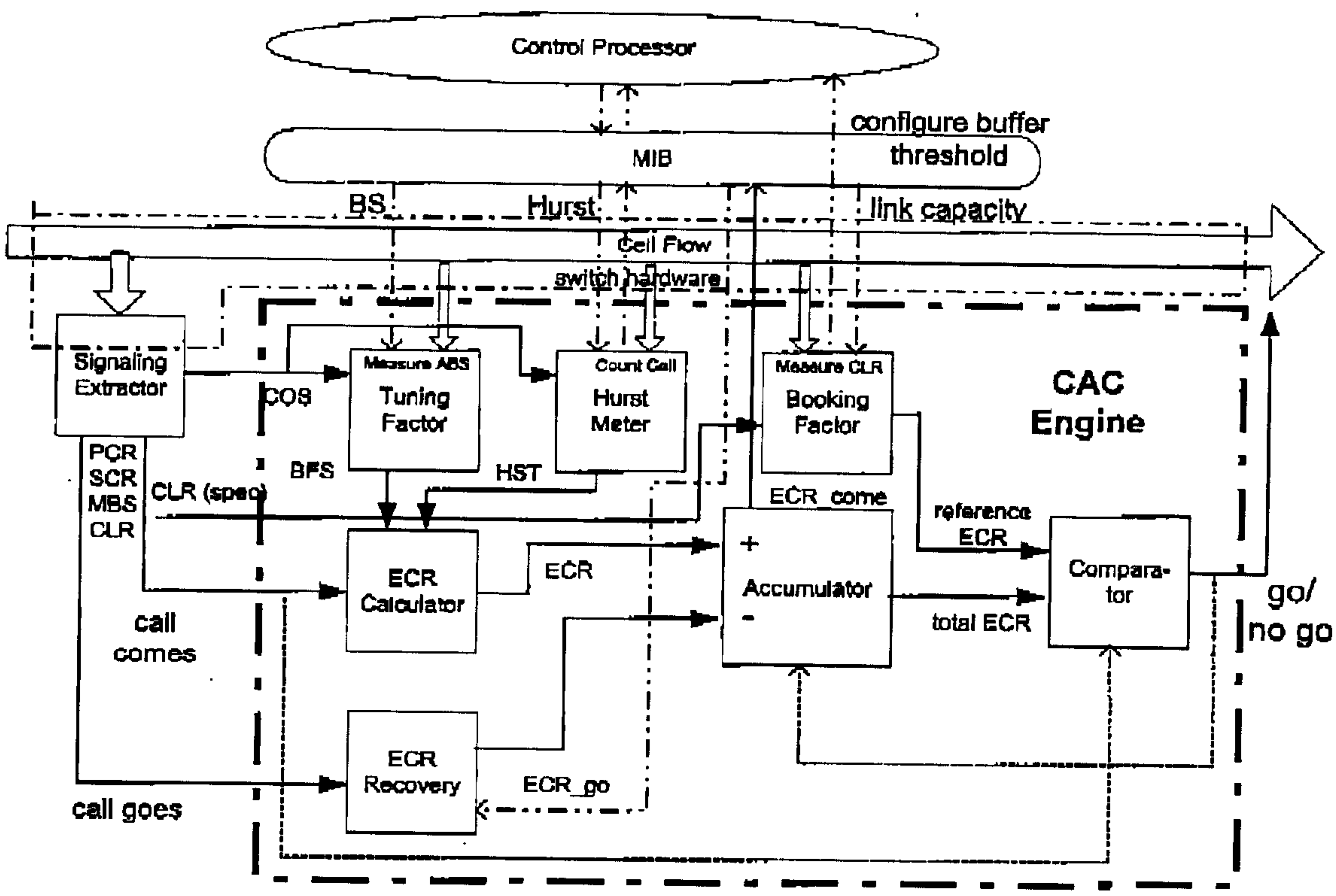


Figure 1

Borden Elliot Scott & Aylen

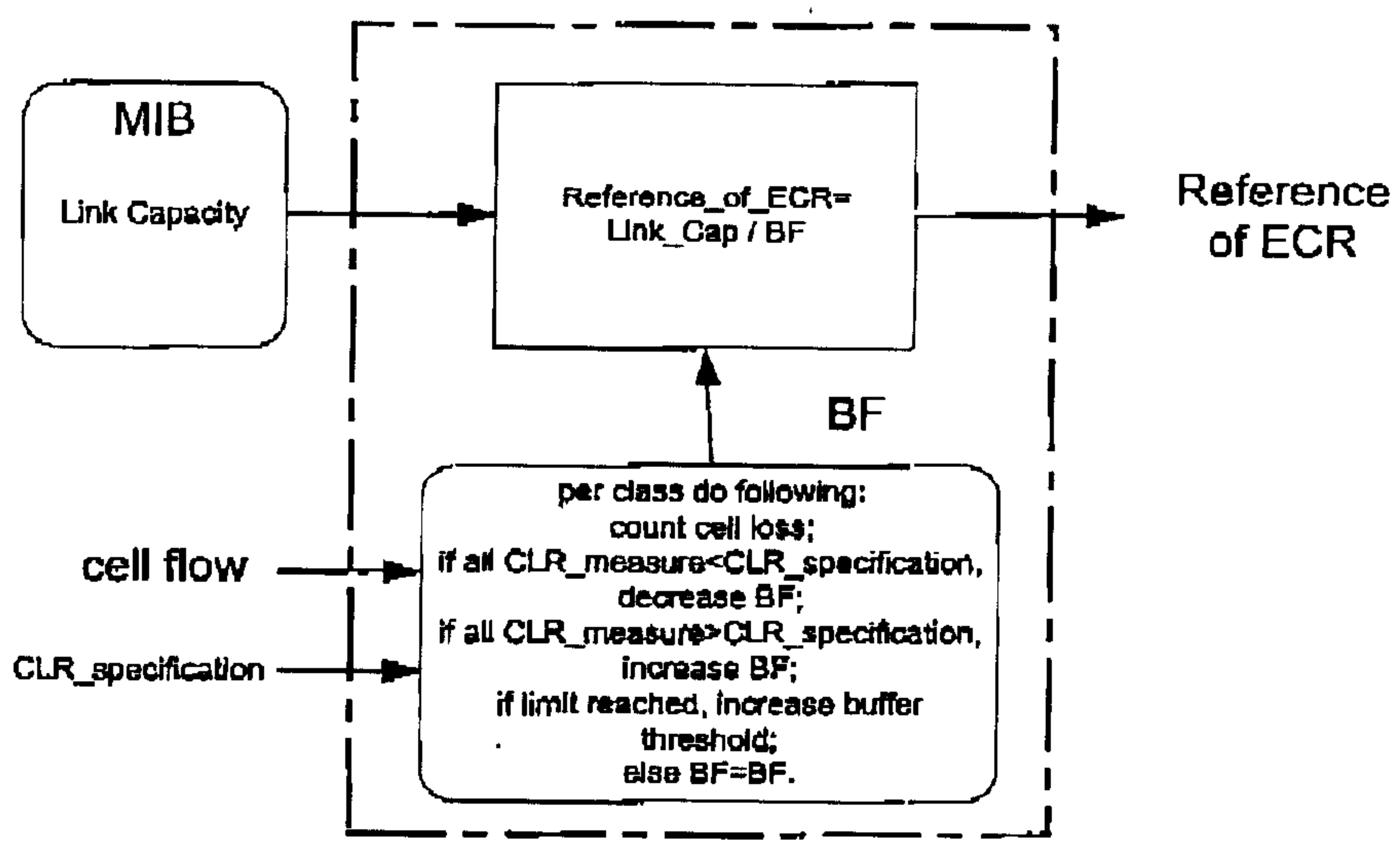


Figure 2

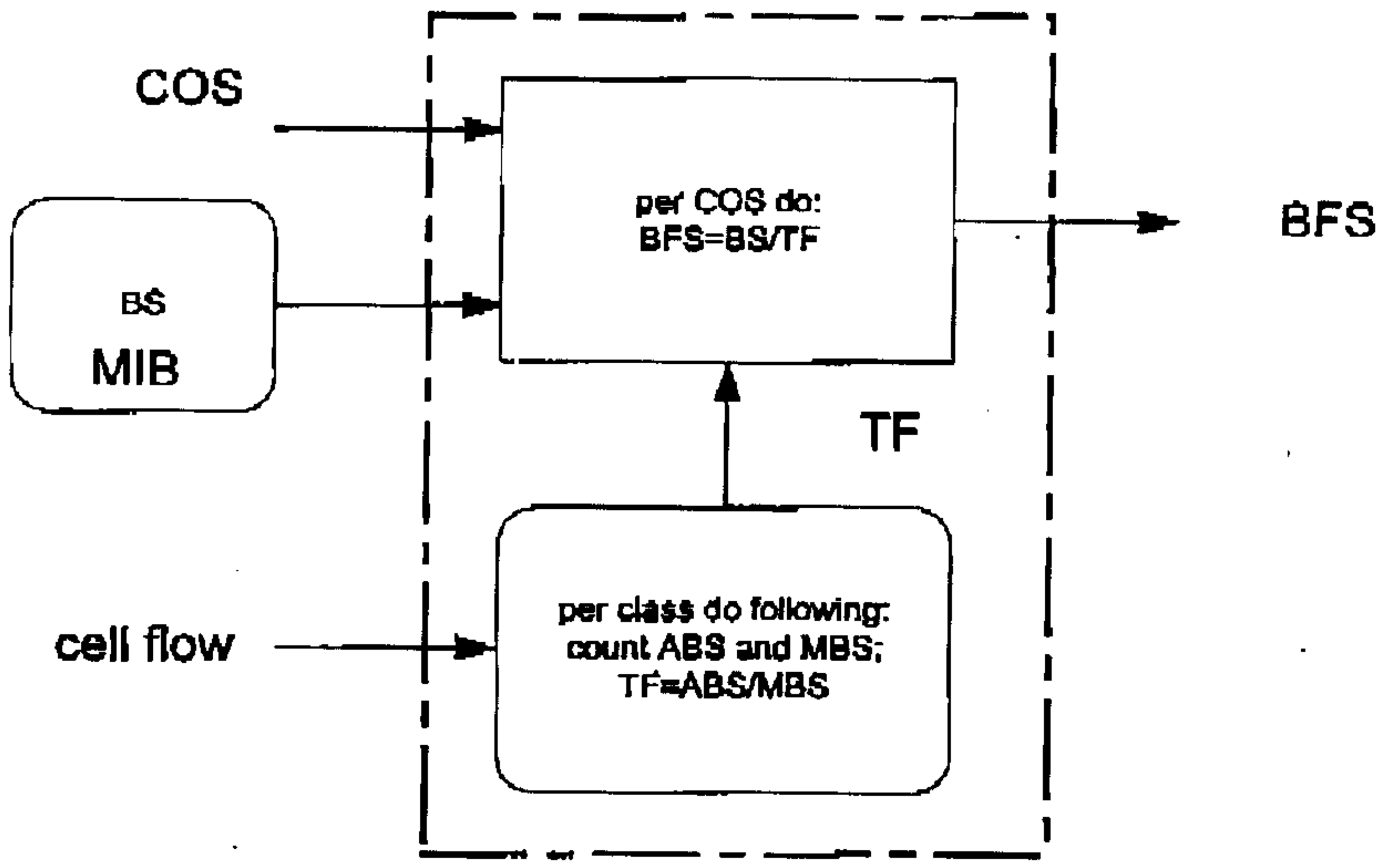


Figure 3

Borden Elliot Scott & Aylen

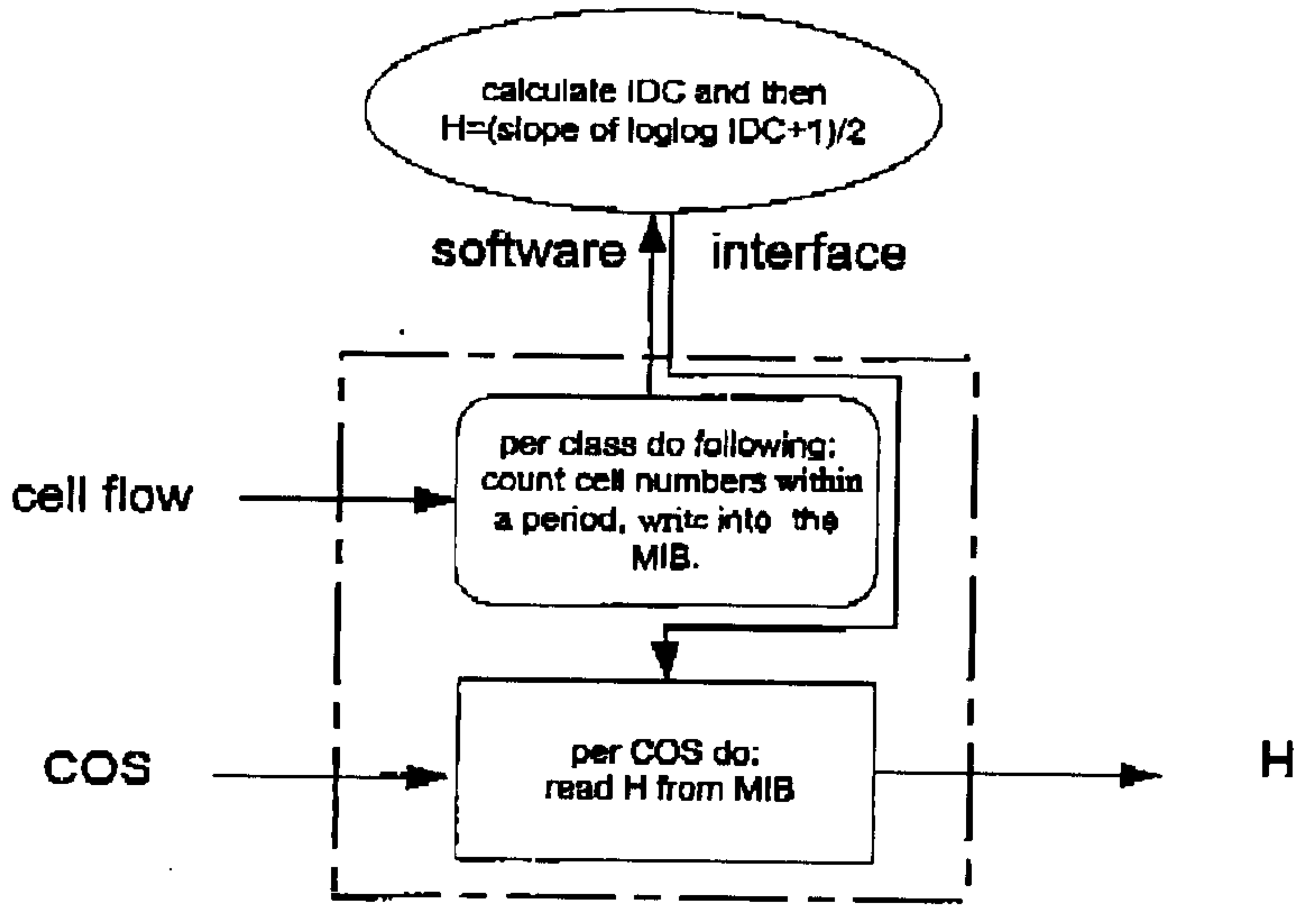


Figure 4

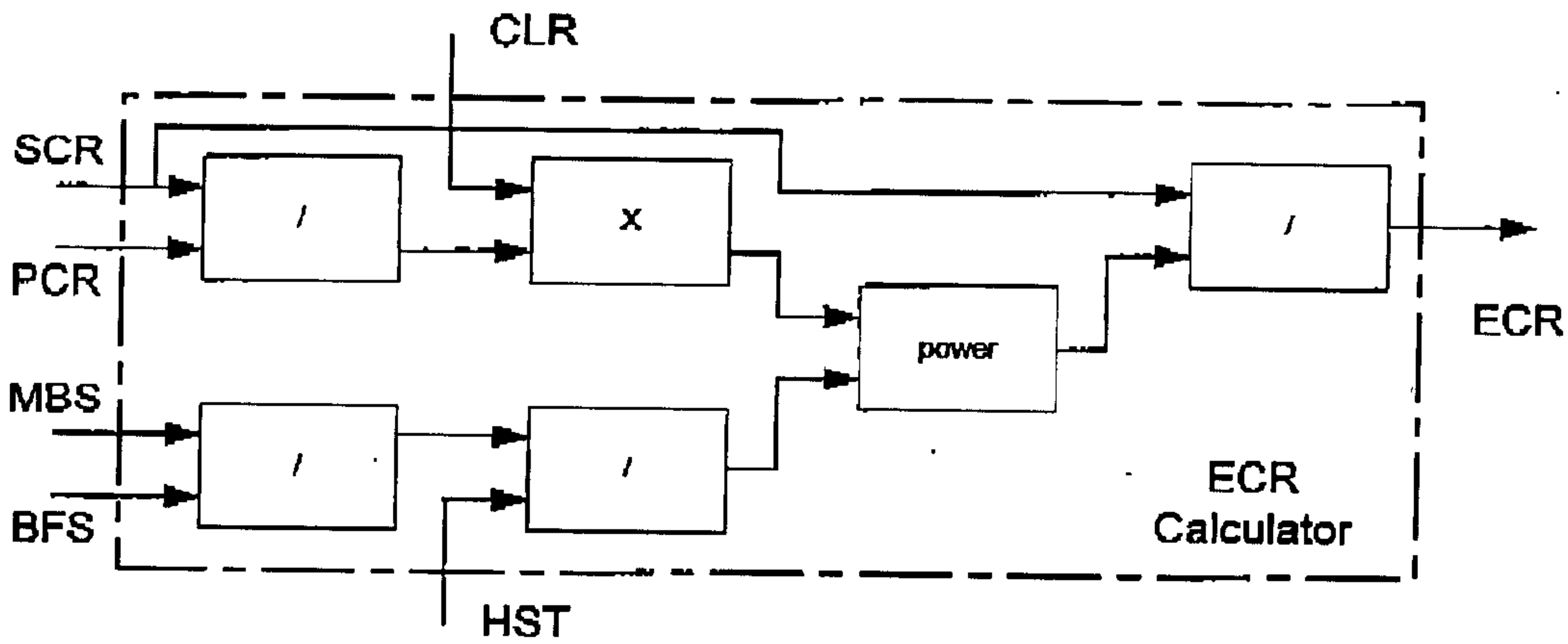


Figure 5

Borden Elliot Scott & Aylen

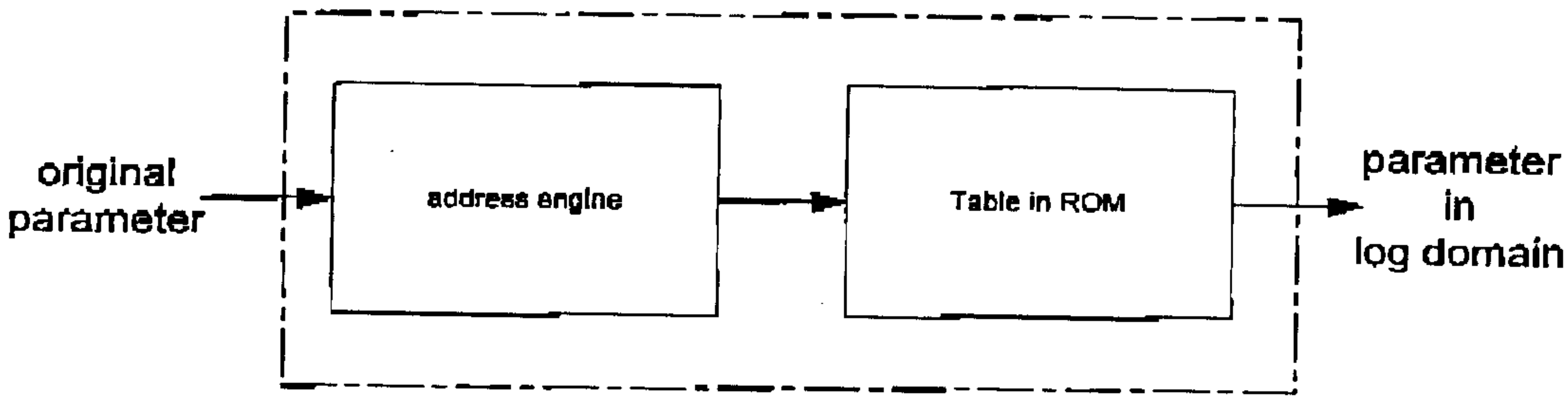


Figure 6

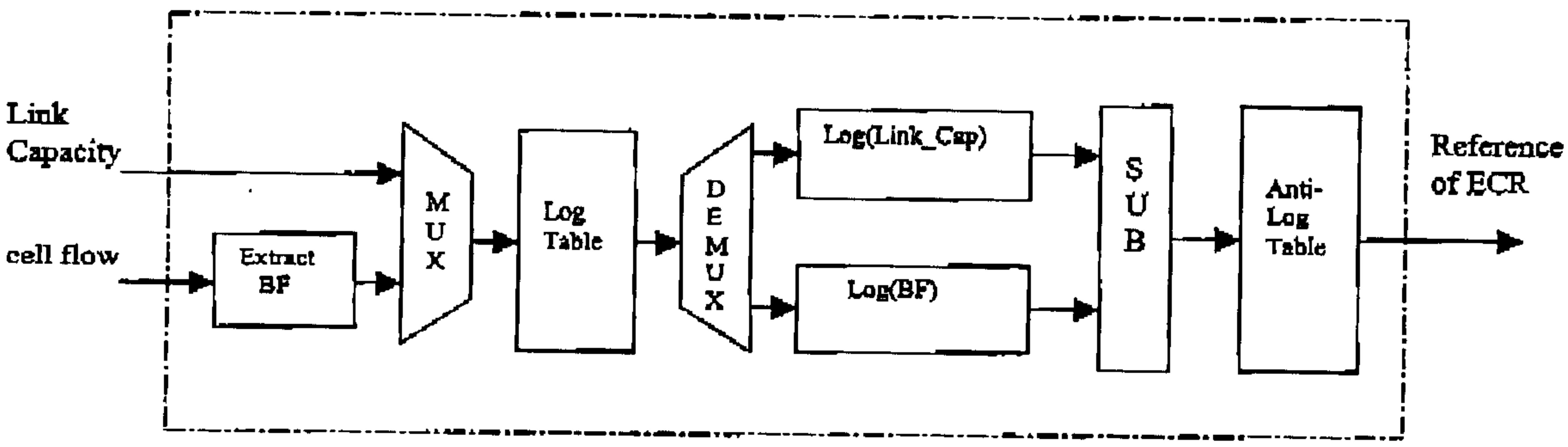


Figure 7

Borden Elliot Scott & Aylen

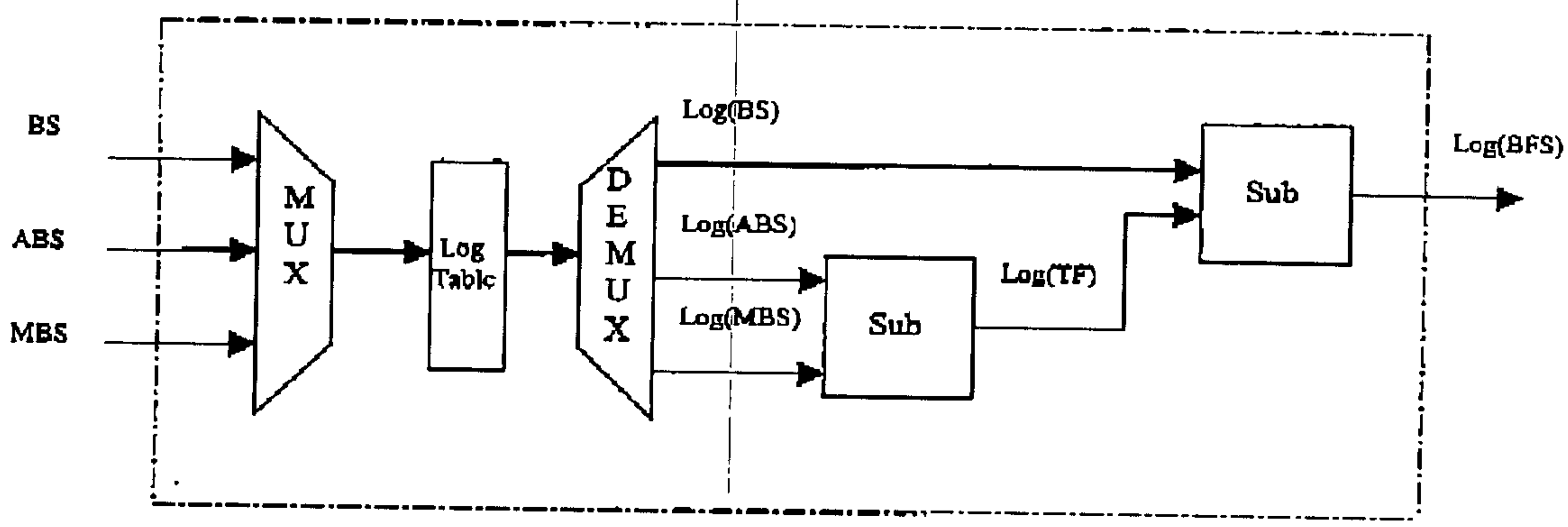


Figure 8

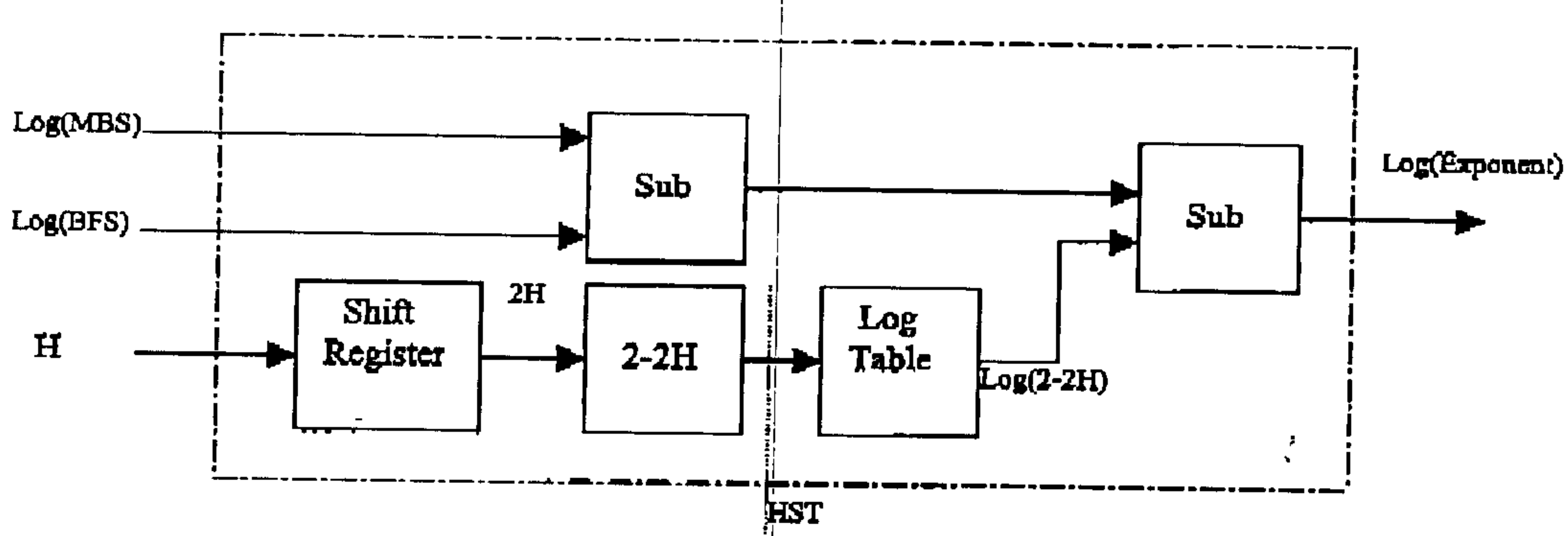


Figure 9

Borden Elliot Scott & Aylen

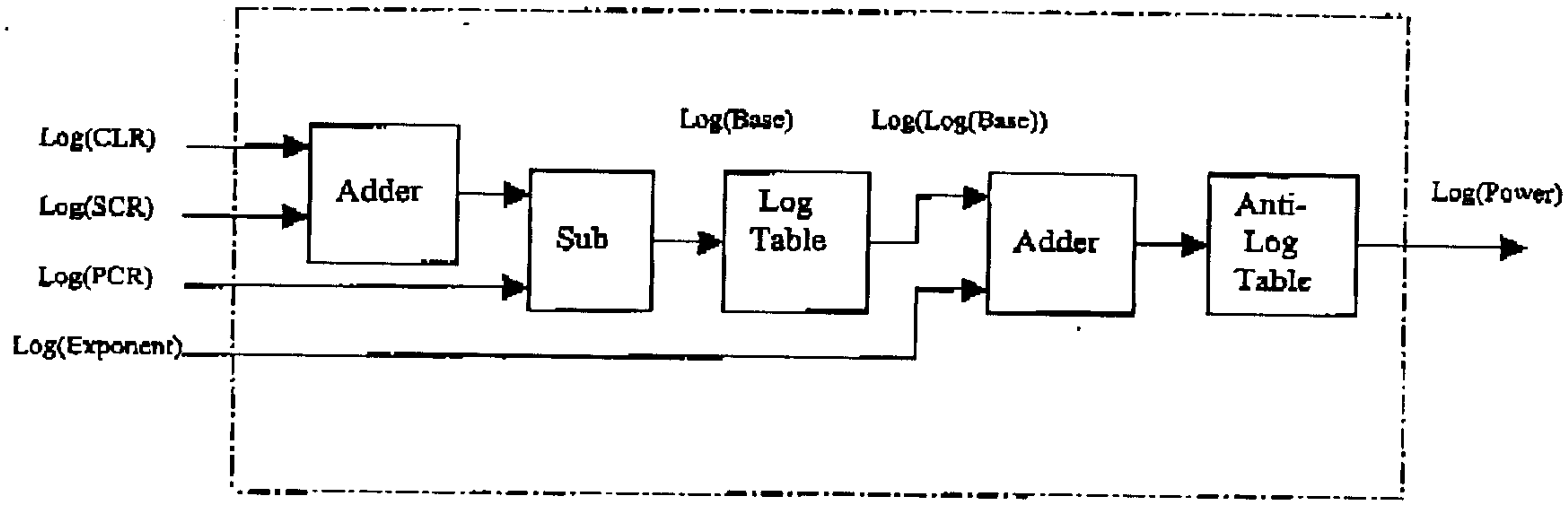


Figure 10

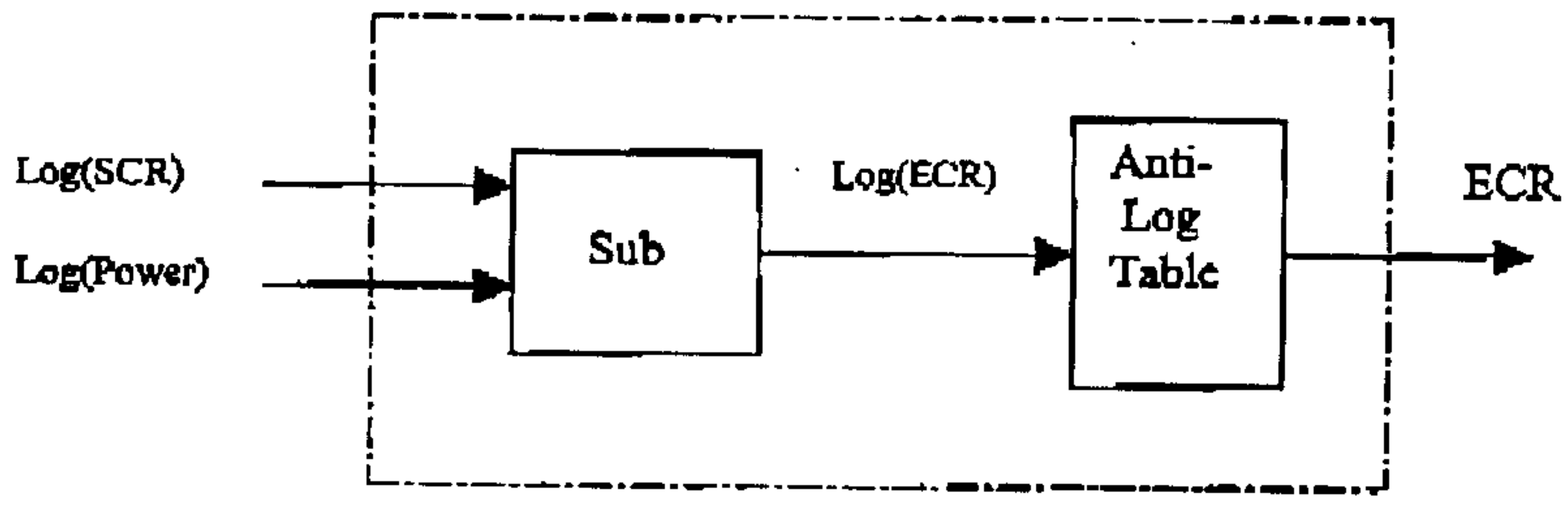


Figure 11

Borden Elliot Scott & Aylen

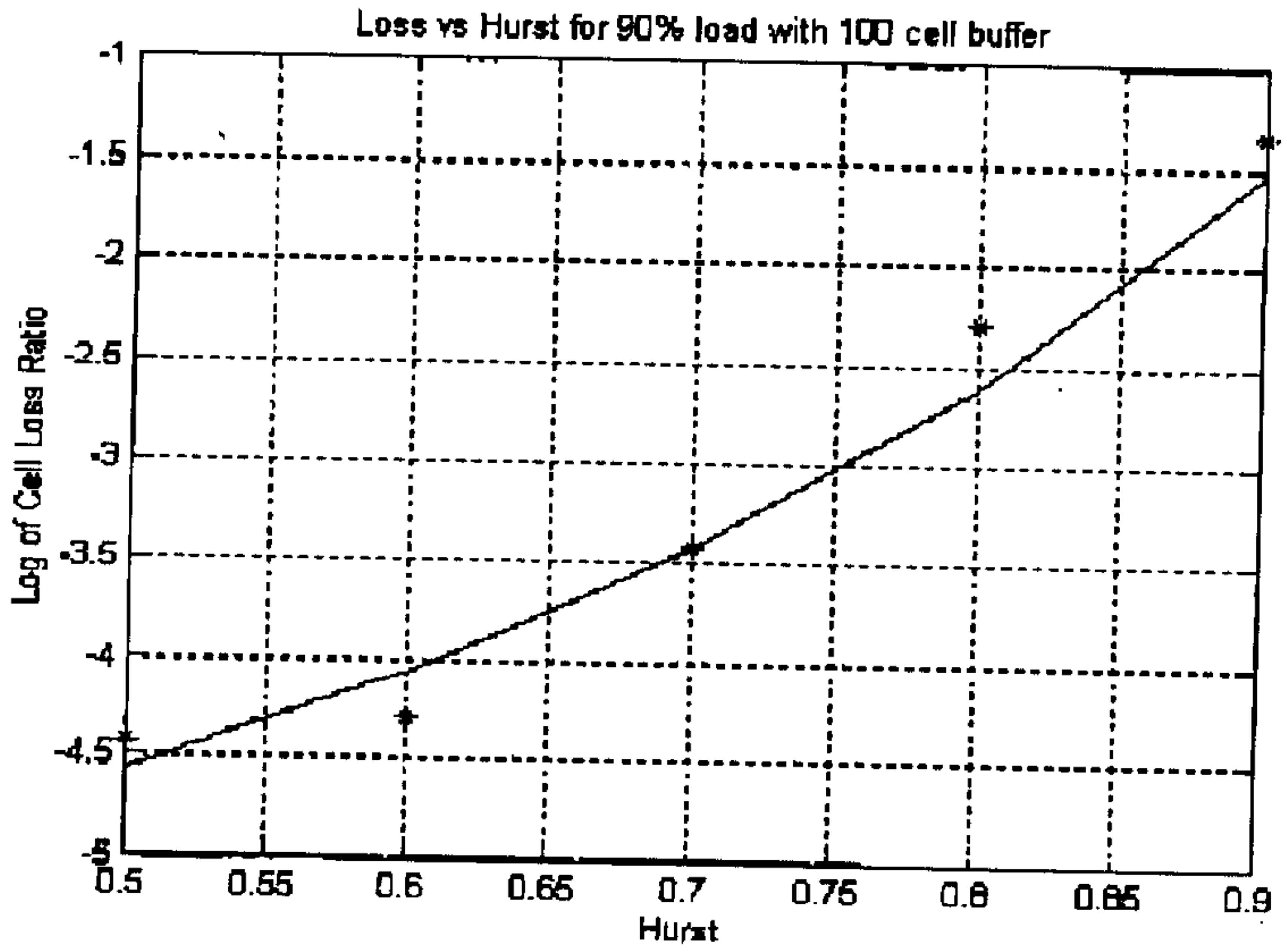


Figure 12

Borden Elliot Scott & Aylen

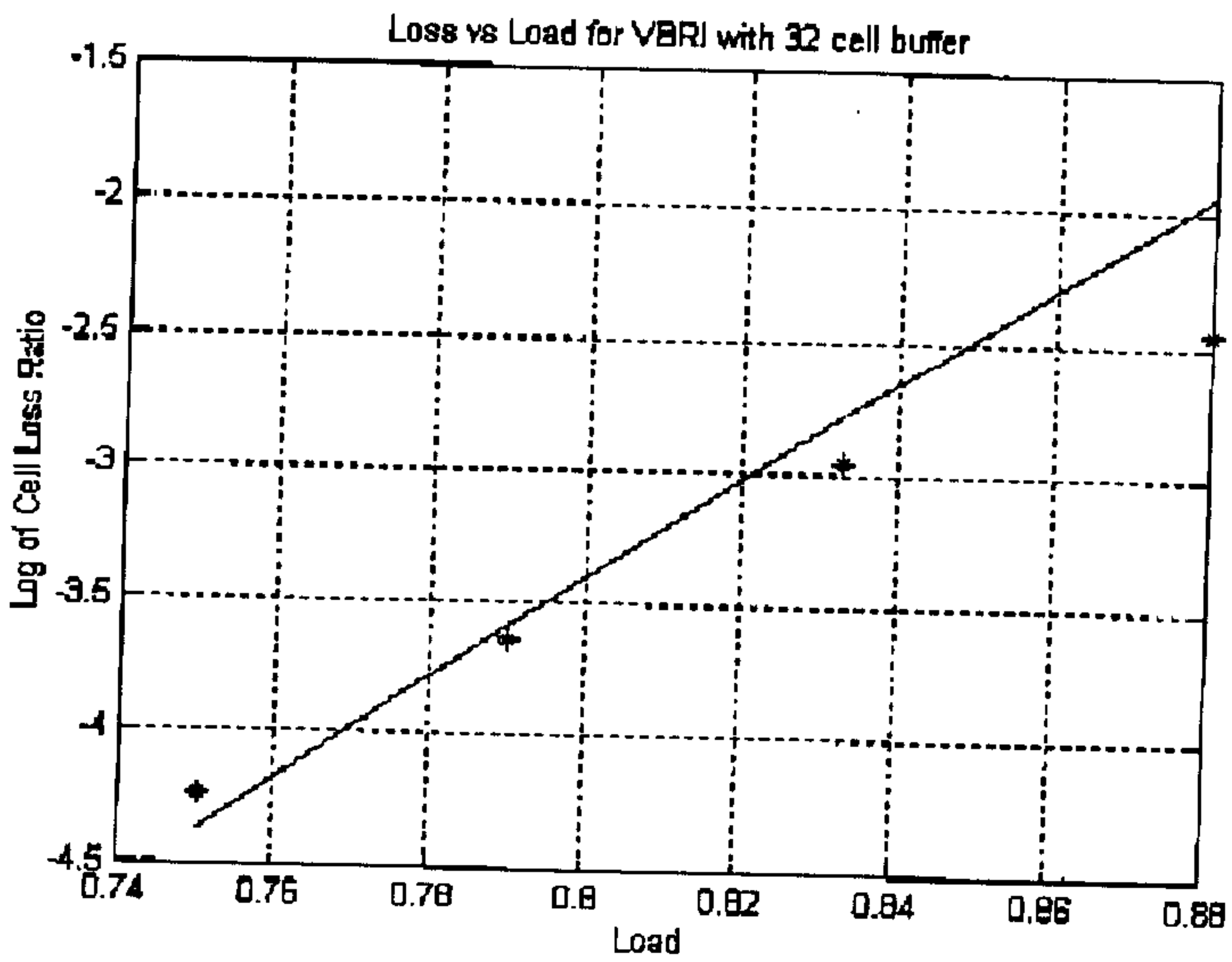


Figure 13

Borden Elliot Scott & Aylen

Number of	"+, -"	"x, /"	power	total
SCAC [Ref.1]	0	7	1	8
MCAC [Ref.2]	8	17	5	30
FCAC [Ref.4]	1	19	7	27

Table 1

ECR (cells/s) of	VBR I	VBR II	VBR III(H)
SCAC	5198	736	10079
MCAC	5271	727	NA
FCAC	NA	NA	10786

Table 2

Number	PCR	SCR	MBS	CLR	BS	BF	TF	H	EB
SCAC	96000	4000	40	1.00E-10	4000	1	1	0.50	5198.3

Table 3a

Number	PCR	SCR	MBS	CLR	BS	r	b	a	EB
ICAC	96000	4000	40	1.00E-10	4000	4.35E-02	4.17E-04	8.81E+02	5271.1

Table 3b

Number	PCR	SCR	MBS	CLR	BS	BF	TF	H	EB
SCAC	96000	640	20	1.00E-10	4000	1	1	0.50	736.3

Table 3c

Number	PCR	SCR	MBS	CLR	BS	r	b	a	EB
ICAC	96000	640	20	1.00E-10	4000	6.71E-03	2.08E-04	4.57E+02	726.8

Table 3d

Borden Elliot Scott & Aylen

Number	PCR	SCR	MBS	CLR	BS	BF	TF	H	EB
SCAC	96000	7439	210	1.00E-06	18867	1	1	0.70	10079.2

Table 3e

Number	PCR	SCR	MBS	CLR	BS	r	b	a	EB
ICAC	96000	7439.554	210	1.00E-06	18867	8.40E-02	2.19E-03	1.15E+03	8540.5

Table 3f

Number	PCR	SCR	MBS	CLR	BS	a	H	EB
VCAC	96000	7439.114	210	1.00E-06	18867	1.03E+04	0.7	10780.6

Table 3g

Borden Elliot Scott & Aylen

load	0.500	0.625	0.714	0.769	0.833	0.909	0.952	0.980	0.990	0.999	0.9999
ratio	2	1.6	1.4	1.3	1.2	1.1	1.05	1.02	1.01	1.001	1.0001
BS_CBRI	15	18	21	23	25	28	30	31	31	32	32
BS_CBRII	11	12	13	14	14	14	16	16	16	16	16
BS_CBRIII	6	6	6	6	6	6	6	6	6	6	6
BS_CBRIV	19	27	35	43	57	85	112	134	143	152	153

Table 4

Borden Elliot Scott & Aylen

Model	CBRI	CBRII	CBRIII	CBRIV
Rate in Cells/sec	4140	16560	119910	173
Initial Jitter in μsec	241.5	60.4	8.34	5780

Table 5

BS	32	64	128	256	512	1024	2048	4096
ratio_VBRI	19.8	15.8	8.5	3.5	1.2	1.1	1.05	1.02
ratio_VBRII	15	18	21	23	25	28	30	31
ratio_VBRIII	11	12	13	14	14	14	16	16

Table 6

Model	VBRI	VBRII	VBRIII
PCR in Cells/sec	58868	14128	353207
MBS in cell	40	20	210
SCR in Cells/sec	2453	94	27372

Table 7

Hurst Parameter	CLR Simulated	CLR Analyzed
0.5	3.8E-5	2.7E-5
0.6	5.1E-5	8.6E-5
0.7	3.7E-4	3.7E-4
0.8	5.1E-3	2.4E-3
0.9	4.6E-2	3.0E-2

Table 8

Borden Elliot Scott & Aylen

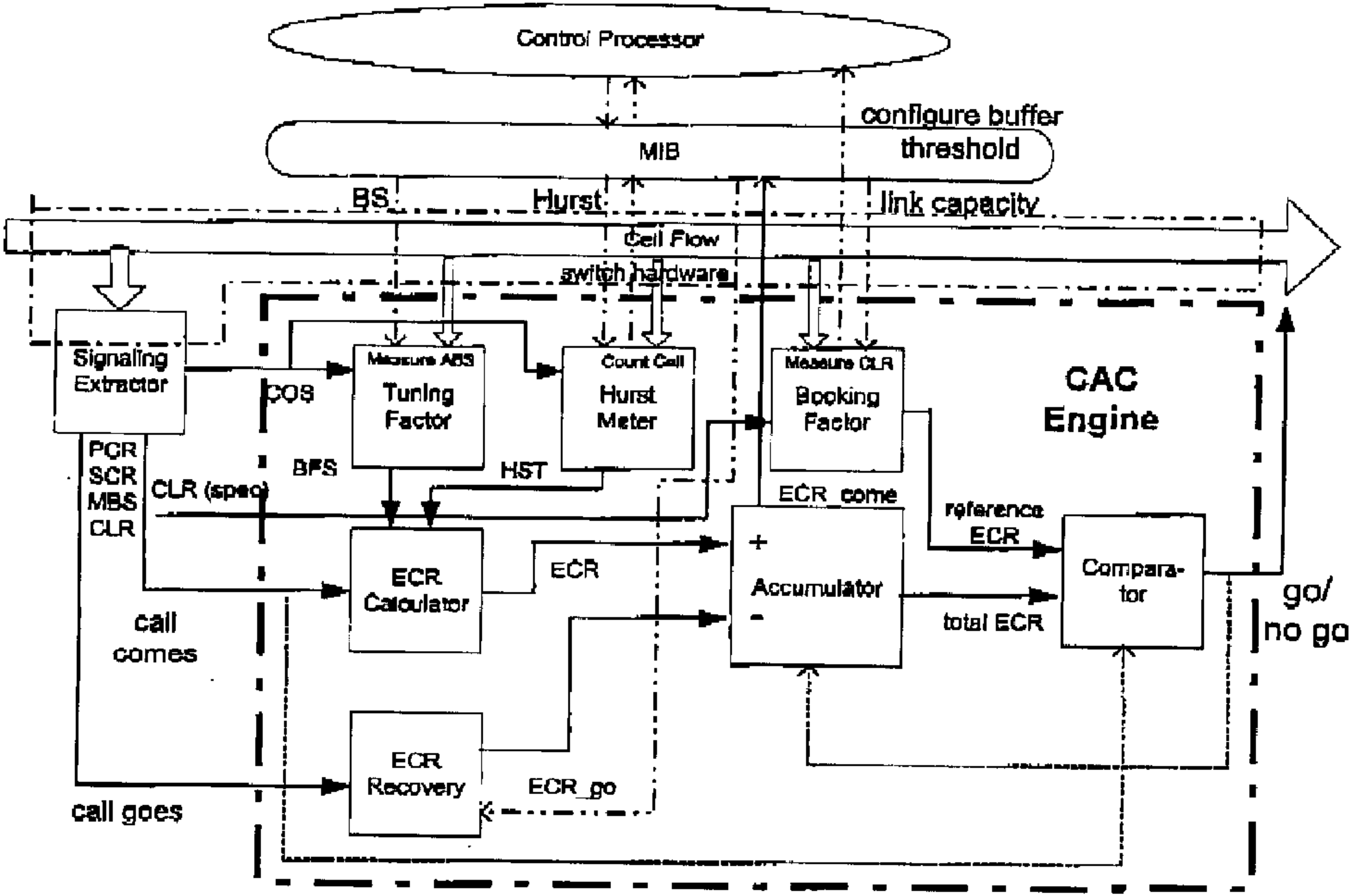
Priority Number	Service Category	Buffered?	Back pressured?
1	CBR	N	N
2	RT-VBR	N	N
3		N	N
4	NRT-VBR	N	N
5		Y	N
6		Y	Y
7	ABR	Y	Y
8	UBR	Y	Y

Table 9a

Priority Number	Service Category	Source	QoS	Buffer Size		ECR/SCR		Fabric CAC
				F	I/E	F	I/E	
1	CBR	CBR	10^{-10}	32	128	1.6	1.05	Y
2	RTVBR1	VBRI	10^{-9}	32	128	21.9	15.7	Y
		VBRII		32	128	123	45	
3	RTVBR2	VBRI	10^{-9}	32	512	21.9	2.6	Y
		VBRII		32	512	123	1.54	
4	NRT-VBR1	VBRI	10^{-7}	32	1712	21.3	1.22	Y
		VBRIII		32	1712	12.6	2.82	
5	NRT-VBR2	VBRI	10^{-6}	32	1712	20.8	1.2	Y
		VBRIII		32	1712	12.6	2.4	
6	NRT-VBR3	VBRI	10^{-5}	32	1712	20.2	1.17	N
		VBRIII		32	1712	12.5	2.1	
7	GFR	TCP	10^{-6}					N
8	UBR	UDP	10^{-4}					N

Table 9b

Borden Elliot Scott & Aylen


Control Processor
MIB
configure buffer
threshold
BS
Hurst
link capacity
Cell Flow
switch hardware
Signaling
Extractor
COS
Measure ABS
Tuning
Factor
Count Cell
Hurst
Meter
Measure CLR
Booking
Factor
CAC
Engine
PCR
SCR
MBS
CLR
BFS
HST
CLR (spec)
ECR_come
reference
ECR
ECR
Calculator
ECR
+
Accumulator
-
Compara-
tor
total ECR
go/
no go
call
comes
ECR
Recovery
ECR_go
call goes