CONGESTION PROBLEM
IN
ATM NETWORKS

Dissertation Submitted to

JAWAHARLAL NEHRU UNIVERSITY

in partial fulfilment of requirements

for the award of the degree of

Master of Technology

in

Computer Science

by

KSHETRIMAYUM BASANTA SINGH

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110 067
January 2000

Lib. Copy

CONGESTION PROBLEM
IN
ATM NETWORKS

Dissertation Submitted to

JAWAHARLAL NEHRU UNIVERSITY

in partial fulfilment of requirements

for the award of the degree of

Master of Technology

in

Computer Science

by

KSHETRIMAYUM BASANTA SINGH

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110 067
January 2000

CERTIFICATE

This is to certify that the dissertation entitled

CONGETION CONTROL
IN
ATM NETWORKS

Which is being submitted by Kshetrimayum Basanta Singh to the **School of Computer and Systems Sciences, Jawahalal Nehru University, New Delhi-110067** for the award of **Master of Technology** in computer sciences, is a record of the bonafide workunder the supervision and guidance of **professor R.C.Phoha.**

This work is original and has not been submitted in the part or full for any University or Institution for the award of any degree.

Prof. C.P.Katti

(Dean SC & SS)

5th January, 2000

Prof. R.C.Phoha 3/1/2000

(supervisor)

ACKNOWLEDGEMENT

I express my deep gratitude to my guide **Prof. R.C.Phoha** for offering constant encouragement, constructive, cooperative, and solutions in helping me for the successful completion of my dissertation.

I would like to record the dean of our school, School of Computer and System Sciences for providing the all efficient facilities in the center for the successful completion of my dissertation.

I take this opportunity to thank all my faculty members and friends for their very good cooperation during the course of this dissertation.

Ksh Basanta Singh 03/11/2000

FROM

Ksetrimayum Basanta Singh

CONTENTS

CHAPTER 1: ATM

1. Objective

The goals of this project are to design and analyze congestion control algorithms for the support of available Bit-Rate (ABR) traffic in the ATM networks. I am studying the behavior of the source algorithm in the rate control scheme. Towards these goals, I describe the problem and present a summary of the overview of the ATM network and its congestion control.

1.1 Asynchronous Transfer Mode (ATM)

ATM is a complex network technology, developed by both telephone companies (Telecom) and computer communication companies (Datacom). The complexity is partly caused by the need for compromises and extra options to support the wishes of both Telecom and Datacom companies. The hope of both is that eventually a global public ATM network will exist like the current Internet. The Global ATM Internet would support both telephone calls and data transmissions using the same basic ATM protocol.

ATM networks are based on virtual connections over a high-bandwidth medium. Despite the complexity, ATM is attracting a lot of interest, because it offers efficient use of high bandwidth, high reliability and Quality of Service (QoS) guarantees. ATM is the network technology chosen for Broadband Integrated Services Digital Network (B-ISDN).

In the next sections, the most important aspects of ATM are pointed out. These aspects make ATM a fundamentally different technology as compared to Ethernet, FDDI and similar ``Layer 2" technologies.

1.1.1 Basic Concepts

ATM is a connection oriented network technology. The connections are not physical end-to-end connections like with telephone networks, but they are virtual connections. This means that, like with telephone connections, a connection must be setup before any data can be transferred between the end-points of the connection. After the connection has been used and is no longer needed, it is terminated (``hanging up the phone").

ATM connections have properties which describe the kind of traffic carried over the connection and the way it is treated by the network. These properties (Traffic Descriptors, QoS parameters, etc.) are negotiated at connection setup time between the calling party, the ATM network and the called party.

As can be expected from a complex network technology, ATM itself consists of a number of layers. The last model is used in figures later in this chapter to indicate the ATM services without going into too much detail.

The ATM protocol stack is best described from the bottom up, starting with the Physical Layer.

Physical Layer

For the transportation of ATM Cells across physical point to point connections (``the wires"), Synchronous Digital Hierarchy (SDH) and Synchronous Optical Network (SONET) protocols are most commonly used. SDH and SONET are very similar protocols, SDH is used mostly in Europe, SONET more in the USA. Physical layer standards are defined by the ATM Forum.

ATM Cells are packaged into the lower layer SONET/SDH frames. At 155 Mbit/s, 45 ATM Cells fit into one frame. Every 125 s, a SONET/SDH frame is sent onto the network, whether it is full or not.

The wires used by these protocols are typically optical fibers or Unshielded Twisted Pair (UTP) copper cables.

ATM Transport Layer

ATM has a fixed transmission unit, called the ``Cell". Each Cell is 53 octets, of which 5 octets are used for the ATM header and the remaining 48 octets can be used for data transport (this part is call the *Payload* or Service Data Unit (SDU)).

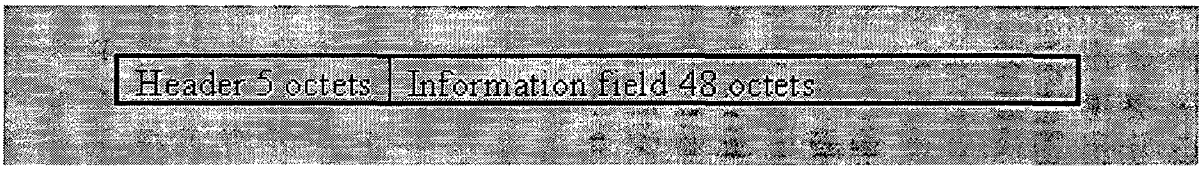


Figure 1.1. (i) ATM Cell Structure

The header can be this small, because instead of the destination ATM address, only the information that is needed locally for data forwarding is included in each Cell. This information consists of a Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) and other administrative data. More on the use of these Virtual Channels and Paths in Section 1.1.2.

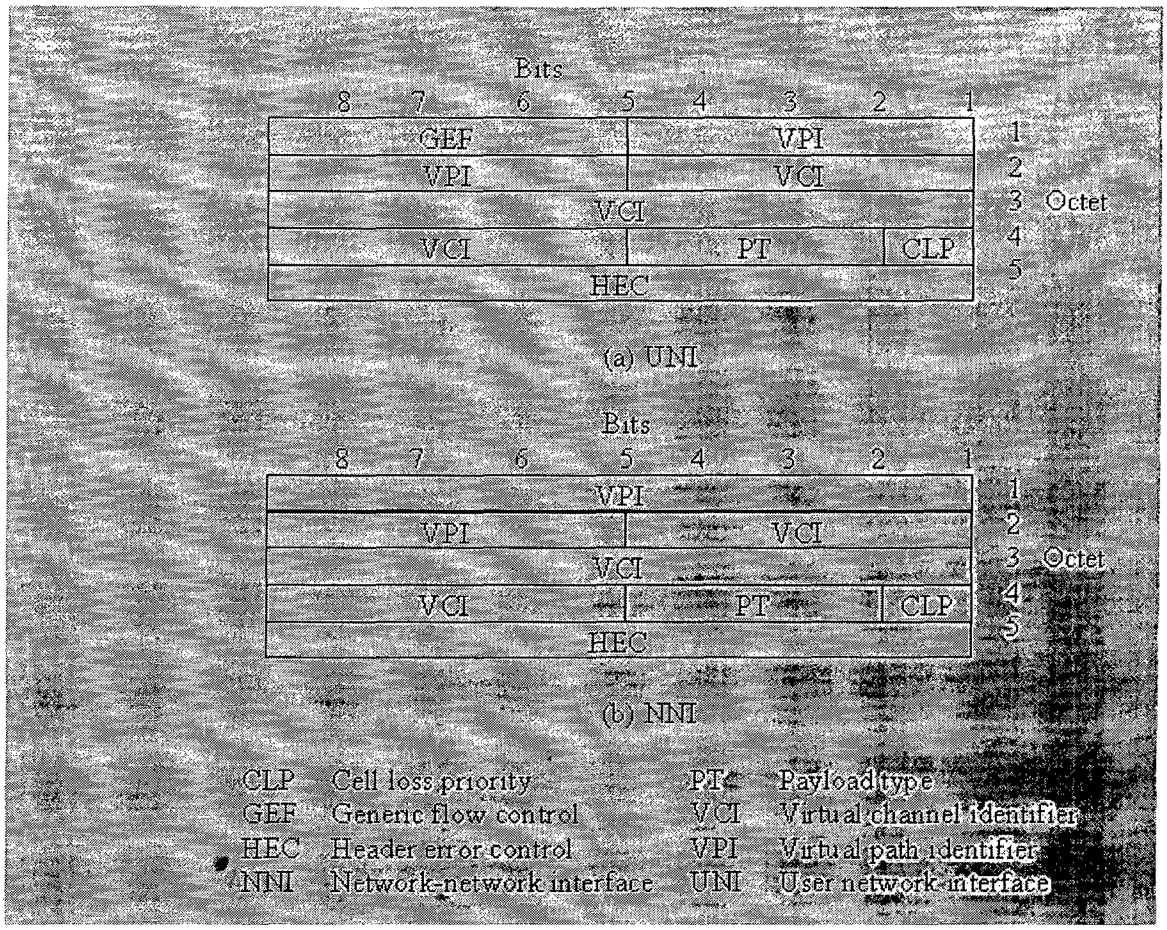


Figure 1.1.1 (ii) Cell header at the (a) UNI and (b) NNI

In the Figure 1.1.1 (ii), the GEF (4 bits) mechanism helps to control the traffic flow ATM connections at the B-ISDN UNI. VCI and VPI discuss in Section 1.1.2. PT (3 bits) field in header used for identification of payload type, cell may contain data, control information for routing, RM (resource management cell) is used for setup the VCC's , in congestion control mechanism. Cell loss priority (CLP) 1 bit field indicates that the cell is high priority if CLP's bit is 1 else 0 for low priority. Header error control 8 bits field used for check over header.

When an ATM connection is set up, the calling party specifies what kind of traffic the connection intends to use and what QoS it expects from the network. This information is put in the traffic descriptor and the QoS parameter respectively and then both are passed as part of the connection setup request to the network, along with other parameters like the destination address. The parameters are carried in *Information* Elements in the signalling messages. The Information Elements are:

The Service Category (a.k.a. Transfer Mode): This is a property of the Virtual Channel Connection (VCC) that is being set up. Currently defined service categories are: Constant Bit-Rate (CBR), (non) real-time Variable Bit-Rate (rt-VBR, nrt-VBR), Available Bit-Rate (ABR) and Unspecified Bit-Rate (UBR). The Service Category defines which QoS parameters should be specified for the connection (e.g. Cell Delay (CD), Cell Delay Variation (CDV) and Cell-Loss Ratio (CLR)). For CBR connections, all three above parameters are performance objectives, but for ABR, only CLR is needed. Traffic Descriptor: Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS). The traffic descriptor is used to allocate resources on the network.

The QoS Class gives the values for the relevant QoS parameters for the chosen Service Category.

After the connection with a certain Service Category has been setup, the ATM client and the ATM network have a traffic contract associated with the connection. If the users of the connection attempt to use too much resources of the network, i.e. exceed the traffic contract parameters, the network may drop cells or use other means within the capabilities of the network to keep the contract. Traffic parameters can be re-negotiated while the connection is still in use. For more information about the information elements used in the signaling messages, please see the UNI specifications and.

When for example a nrt-VBR connection with a certain PCR, MBS and SCR has been setup, the connection users do not constantly supply ATM Cells at the PCR. The network may allocate more bandwidth on a link than the sum of all Peak Cell Rates of the VCC that go through that link. This is called statistical multiplexing. Statistical, because it uses statistics to calculate the chance that all connections need to use their connection at PCR at the same time. The assumption is that with enough simultaneous connections open and as long as the traffic is not related to each other, the link's bandwidth usage can be described by statistical methods.

ATM Adaptation Layers

While at the ATM level, traffic in all service categories can be transported, higher layers may want more specific services, better suited to the kind of traffic produced by those layers. E.g. not all B-ISDN users need a timing relation between the sending and receiving entities. For this purpose, the ATM Adaptation Layers have been defined. When new kinds of traffic are recognized, a new AAL can be defined for it, while the ATM layer remains unchanged.

For the purpose of transporting connectionless traffic as produced by the IP layer, AAL5 is best suited. Although AAL5 is not generally considered to be a Class D service category, it is chosen for the low amount of overhead in the AAL5 PDU and the simplicity of its implementation. Usually IP is carried over a connection with service class ABR or UBR, but nrt-VBR would be suitable as well.

The AAL takes care of the Segmentation into, and Re-assembly of the ATM Cells. Additionally, an AAL may check for cell-loss, timing or data errors and perform multiplexing of higher layer data streams into a single ATM Virtual Channel Connection.

AAL 3/4 segmentation and re-assembly protocol data unit message format (SAR-PDU) have 10 bits MID (multiplexing identifier) field, it may support multiplexing, The MID field assists in the interleaving the cells from different CS-PDU(CS -Convergence sub-layer) and re-assembly of these CS-PDUs.

Adaptation Layer 5 is the most commonly used AAL, it provides an un-guaranteed connection oriented transport of variable-length frames with error detection. The user data frames can be up to 65,535 bytes, which gets segmented into 48 byte chunks, after adding AAL 5 control information and padding. These chunks are sent as payload in the ATM Cells. Both CLIP and LANE use AAL 5 to transport their protocol frames across the ATM network.

ATM Signaling Entities

For the setup, modification and release of the connections and other signaling between the ATM network (the ATM switches) and the ATM end nodes, signaling entities are defined. These entities use the special Signaling ATM Adaptation Layer (SAAL), to encapsulate their signaling data in ATM Cells. Since signaling is used to set up Switched Virtual Connections(SVCs, see Section 1.1.2), a Permanent Virtual Connection, identified by VPI=0 and VCI=5, is used for exchanging signaling information.

ATM Bearer Service

Later on, we will see the ATM control and user plane used as a single service provider. The combination of the signaling and transmission services will be called the `` ATM Bearer Service" (see the simplified stack on the right in Figure 1.1). It will provide the service of setting up a connection (the VCC), transmitting data in AAL packets over the ATM VCCs and closing down the connection.

1.1.2 Network Structure

ATM networks consist of ATM switches connected by point-to-point ATM links (see Figure 1.2). ATM hosts are connected to the local switches. There are two types of ATM Networks; the Private Network and the Public Network.

The ATM nodes (hosts or switches) use the Public User-Network Interface (Public UNI) protocol. When hosts are connected to a local switch the Private UNI is used. Switches within a private network use the P-NNI (Private Network-Network Interface or Private Network-Node Interface). Switches within the public network do not use the P-NNI, but eventually a derivation of it will probably be used there as well. For the implementation of an IP multicast capability, the Private UNI at the ATM host side is most important.

Virtual Connections

ATM, being a connection oriented network technology, provides its service users with the possibility of setting up a Virtual Channel Connection (VCC) with a certain traffic contract. The VCC is a direct connection between the end-points (to the protocol layer above the ATM Transport Layer) across the ATM network. Once this connection is set up, a VCC is identified at the end-points by a Virtual Path Identifier (VPI) and Virtual

Channel Identifier (VCI).

The ATM transport layer itself is built up of two layers; Virtual Channel and Virtual Path. At the service interface of the Virtual Path Service Provider a Virtual Path Connection (VPC) can be setup (with traffic and QoS parameters), identified by a VPI. A VPC can carry multiple VCCs, which have the same VPI but different VCIs.

In general, ATM hosts are connected to each other via ATM switches, as can be seen in Figure 1.2. A VCC between two ATM hosts goes through one or more switches in the network. The VPI/VCI combination that identifies the VCC at the end-points has only local significance. The switches translate either only the VPI (a VP-switch) or both the VPI and the VCI (a VC switch). The section of a VPC between two VP Switching Entities is called a Virtual Path Link (VPL). The section of a VCC between two VC Switching Entities is called a Virtual Channel Link (VCL). The VCI retains its significance through a VP switch, so VCLs are carried through VP switches. A decomposition of a VCC in VPLs and VCLs is shown in Figure 1.3, the point where a VPI or VCI gets translated is shown with an arrow. Note that the Virtual Channel Connections terminate at the service interface of the ATM Transport service provider.

ATM switches have `` cross-connect tables" to translate the VPI/VCI connection (or rather, link) identifiers in the incoming ATM Cells to VPI/VCI combinations in outgoing ATM Cells. This translation can be done in hardware once a connection has been setup and should be much quicker than the IP routing mechanism.

The cross-connect tables can be filled by the network management or by signaling protocols. A connection set up by management is called a permanent connection. This can be either a Permanent Virtual Path Connection (PVPC) if only VP switching entity's cross-connect table is configured, or a Permanent Virtual Channel Connection if the VC switching entity's cross-connect table is configured. When signaling is used to setup a VPC or a VCC, the connection is called Switched VPC (SVPC) or Switched VCC (SVCC) to signify that it was a connection established through signaling commands, rather than by management interference. The connections themselves, permanent or switched, are exactly the same, regardless of how they have been established.

This concept of using (virtual) connections for communications is very different from the way IP packets are sent and routed. When implementing IP on top of ATM, the IP entity does not care if the VCC transporting its packets is released when it is not used, the when and how of opening and closing of connections for communications is a matter of optimization.

Native ATM addresses

ATM users communicate using virtual connections, identified by a VPI/VCI pair, during the data transport phase. In order to setup a connection, the switches need some way to identify the called party, so the routing protocols can find a route to it during the connection setup phase. ATM addresses have been defined, so the signaling entities can address an ATM end-point to their peer signaling entities in the ATM network. These addresses are modeled after the OSI Network Service Access Point (NSAP), as defined in ISO 8348 and ITU-T X.213. In Figure 1.4 the three types of ATM addresses are shown. The Data Country Codes (DCC) are specified in ISO 3166. The International Code Designator (ICD) is maintained by the British Standards Institute. The E.164 format is the ISDN number format. All three formats are 20 octets long .

The address consists of 3 parts, the Authority and Format Identifier (AFI), the Initial Domain Identifier (IDI) and the Domain Specific Part (DSP). The AFI part determines the rest of the address interpretation. The DSP consists of a High Order DSP (HO-DSP), an End-System Identifier (ESI) and the selector byte (SEL). The SEL-part is never used by the routing algorithms, which means that each ATM end-point can use 256 different ATM addresses.

ATM end-points are not configured with a complete ATM address, the address is built up at initialization time across the UNI. For this purpose the Interim Local Management Interface (ILMI) is used to perform address registration procedures. The ATM end-point supplies the ESI and the SEL part of the address as ``user part", the ATM network supplies the rest of the fields as ``network prefix".

The interconnected structure of these networks is such that it is usually possible to choose from several routes between the two hosts. There are some options in the internet protocol to help select the best route. These options specify certain qualities of the network, like high throughput or high reliability. The routers may or may not have the ability to use these options. When a network or IP-router in one route to the destination host is not operational, an intermediate router can chose another route to reach the same destination IP-host.

1.1.3 User Network Interface

The User Network Interface is used by ATM hosts to communicate with ATM networks and by private ATM networks to communicate with public ATM networks. All aspects of the User-Network Interface, from the lowest physical details to the signaling messages exchanged above ATM level, are specified in the ATM standards documentation of UNI 3.1. The signaling part of the UNI has been updated in version 4.0.

Signaling UNI

UNI 3.1 specifies a number of capabilities that are supported by ``Phase 1" signaling. These capabilities include support for point-to-point connections and point-to-multipoint connections. It also provides protocol support for the basic signaling functions to establish, maintain and release these connections.

Point-to-multipoint VCC support is limited to unidirectional VCCs (the return path is defined to have zero bandwidth, PCR=0) from the root to the leaves of the multicast tree. Only the root may add and remove leaves, which causes the use of ATM multicast to be difficult to scale up in number of leaves. This is because the root node must handle more requests to be joined and exchange connection messages with the network to add and remove leaves. The root node is the bottleneck in the add/leave procedure.

In UNI 4.0 the point-to-multipoint connection signaling is extended to support Leaf Initiated Join, either with or without root-node intervention. Especially the latter would make ATM point-to-multipoint connections significantly more scalable. Also new in UNI 4.0 is an `` Any-cast" service to setup a point-to-point connection to any one of a group of ATM end-points. This last feature would allow load sharing and redundant services without bothering ATM clients with connection failures.

In the UNI 3.1 and UNI 4.0 signaling specifications, no support for multipoint-to-multipoint connections is defined. In the signaling specification of UNI 3.1, two possible solutions to this problem are given. To support multipoint-to-multipoint connections, a Multicast Server (MCS) can be defined. The MCS connects to all recipients using a point-to-multipoint connection. The senders connect to the MCS using a point-to-point connection when they want to transmit to the members of the multicast group. Alternatively, all sending ATM end-points can set up point-to-multipoint connections to all receiving end-points.

CHAPTER 2: RATE-BASED ACCESS CONTROL FOR ATM NETWORS

1. Introduction:

The statistical multiplexing of various kinds of sources with diverse traffic characteristics in the ATM networks leads to serious congestion problems. These problems, which are characteristics by cell losses and excessive delays, make bandwidth allocation ineffective without control. One of the most important of them is the policing of sources. This suggests that, a bandwidth enforcement mechanism should be specified to control source traffic parameters negotiated during the call setup phase, and limit them when violate the contract. This policing mechanisms should operate in real time, be simple, fast and cost effective to implement in hardware.

Several traffic enforcement mechanisms have been purposed so far. A brief review of them can be found in later conditions. Most of them try to estimate the traffic parameters of a source in order to control it. This is the reason why they need long observation periods. On the other hand, especially in the case of bursty sources, the statistical nature of burstiness leads to ineffective estimation and thus. Ineffective control, particularly when the estimation decreases. So, the attempt to control to the source traffic parameters conflict with the need for a good estimation (which implies long observation periods) and the time required to detect unwanted changes (which implies that the observation periods should be short). Consequently, this policing mechanisms are not able to control the traffic parameters of a source effectively and quickly, especially as far as the mean rate and the mean burst duration are concerned.

However, a learning algorithm could "learn" the behavior of a source. In the proposed scheme, a Stochastic Estimator Learning Algorithm(SELA) is employed to enhance the Leaky Bucket(LB) in order to police bursty sources effectively. SELA tries to find out whether the source transmits according to its negotiated traffic parameters and ,if it does not, limits it to the negotiated values. Th performance of the proposed mechanism on a bursty source, as well as its effect on an inter-nodal node where many policed sources share the same buffer, were investigated through simulation. As will be shown, this new scheme controls the non-conforming sources more tightly than the LB. Thus, it achieves more statistical gain (saves more bandwidth) and guarantees the Quality of Service (QoS) constraints better, while the average reaction time is shorter. The proposed methodology has no computational overhead, since it requires only few computations, and it can be implemented easily in hardware.

This chapter is organised as follows : in the next section , after describing the considered traffic models , a brief review of the most known policing mechanisms is given , while the LB mechanism and related problems are extensively analysed. Moreover ,some concepts of Learning Automata theory are introduced and SELA is presented. In the next of the above , it describes fluid flow model of LB based on the proposed mechanism is presented. After this numerical results obtained from simulation and a comparative performance study are presented. Finally , in the last of the chapter it presents the conclusions and plans for the future work are discussed.

2. **General concepts:**

2.1 Traffic model

In this small chapter of bursty sources which consist of active(ON) and idle(OFF) states, such as, packetised voice, high speed data and still picture sources are employed. Several papers in0 the literature have used the state Markov chain successfully to model such sources. Both active and silent periods are assumed to be exponentially distributed with averages T_{on} and T_{off} respectively .A source can be characterised by the following set of parameters:

- p: peak(burst0bit rate (in Mb/sec)
- m: mean bit rate (in Mb/sec)
- L: average burst length (in cells)
- T_{on}: mean burst duration (in sec)
- T_{off}: mean silence duration (in sec)
- b: source burstiness

where b=p/m, $T_{on}=L \cdot n_{cell}/p$($n_{cell}$=53bytes=424 bits is the cell length) and $T_{off}=T_{on} \cdot (b-1)$

2.2 A brief review of related work:

Several policing mechanisms have been proposed so far , such as the "Leaky Bucket" (LB),the " Jumping Windows"(JW) , the "Triggered Jumping Windows"(TJW)the "Moving Windows"(MW) and the "Exponentially Weighted Moving Average" (EWMA). The Jumping Window mechanism imposes an upper bound m on the number of cells which are permitted to enter the network during the time interval T (called window). Each window commences immediately after the end of the preceding one. All consecutive cells arriving after the first m cells during a window are dropped or marked as excess cells. In the Triggered Jumping Window mechanism , the window starts with the first arriving cell of a new burst. Thus, in this case consecutive windows are not necessarily consecutive in time. The Moving Window scheme differs from the Jumping Window in the fat that the window is steadily moving along the fact that the time axis with the result that each cell is taken into account for the duration of one window. Finally, the main difference between the Exponentially Weighted Moving Average and the Jumping Windows is that in the EWMA scheme the upper bound of the number of accepted cells during a window is not constant, but , instead it is an exponentially weighted sum of the number of the accepted cells in the preceding window and the mean number of cells. An extensive presentation of the Leaky Bucket mechanisms follows in the later section. It is generally agreed that , the LB performs better than the above ones. The problem described in the introduction that these mechanisms encounter has led to use of artificial intelligence techniques. It used another method called Artificial Neural Networks(ANNs) which tries to police the probability density function of the source . it uses two ANNs which are trained off-line for possible cases of a source behavior, while during the on-line operation their outputs are compared to determine whether the source is a conforming one or not. However, there is no guarantee of the convergence on the correct decision, while the size of the ANN increases as the number of the possible violations that ANN should be trained for increases . finally , a policing mechanism based on fuzzy logic was proposed. This scheme , although it achieves detection of very small deviations from the negotiated values, can not be used for the short-term fluctuations, while it can police only one source traffic parameter at a time. This fact, in combination with its increased hardware complexity and the necessary tuning of the mechanism's parameters, render it a rather complex and costly policer.

2.3 The Leaky Bucket mechanism:

There are two main versions of the LB mechanism. In the simpler one, the LB consists of a counter that its value increments by one each time a cell is transmitted by the source and decrements leaks according to a suitable rate r as long as the counter value is positive. When the counter has reached a given threshold Q, an arriving cell is dropped or marked as an excess cell.

The second version of the LB is presented in the figure given below:

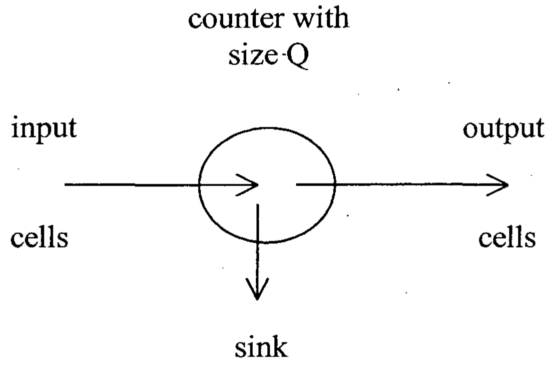


fig: first version of Leaky Bucket.

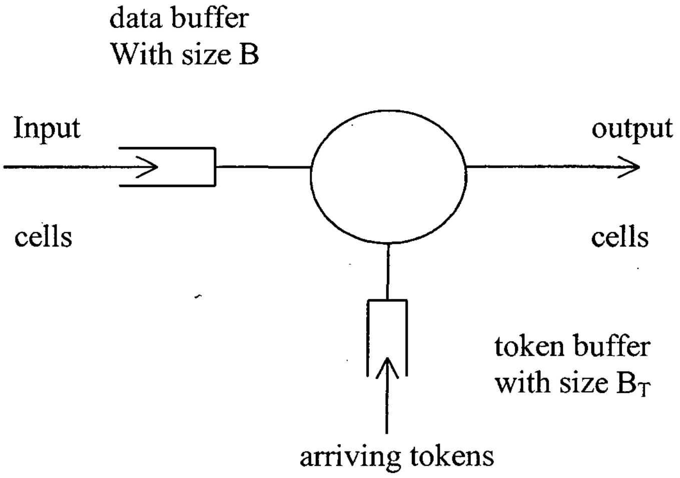


fig: second version of the Leaky Bucket

In the figure where instead of a counter there is a token pool with size B_T and a data cell buffer with size B_D . Tokens arrive at the token pool according to a constant rate γ. Whenever the user transmits a cell, a token is consumed from the token pool. Arriving tokens which find the token pool full are dropped. If the tokens pool is empty, no cells are allowed to be transmitted. The cell loss probability in the second version depends on B_D and B_T only through their sum. The data buffer allows excess cells to wait resulting in a smoothed cell arrival process at the network. The partitioning of B_T and B_D determines the trade-off between the cell loss probability and the cell delay. If B is fixed, small B_T and consequently large B_D results in the smooth outgoing traffic with small cell loss probability and big delays.

Without the data buffer $B_D=0$, all the differences between the two versions vanish. The threshold of the counter Q coincides with the size of the token pool B_T and r coincides with γ.

2.4 Why the Leaky Bucket does not work?

The source traffic parameters that a policing mechanism should enforce are usually the peak rate, the mean rate and the mean burst duration or the maximum allowed burst duration. Evidently the peak rate of the source can be controlled easily and effectively. Several mechanisms have been proposed for example, dropping cells when their inter-arrival time is smaller than the reciprocal of the negotiated peak rate.

The source mean rate is an essential parameter for many admission control schemes. Thus, its efficient control is very important for a policer. In the case, if we define m_o as the negotiated mean rate and P_{QoS} ,the cell loss probability as specified by the negotiated QoS requirements, then the ideal behaviour of a policer should be:

- In case of $m<=m_o$, $p_{loss}<p_{QoS}$,
- In case $m>m_o$,$p_{loss}=p_{des}$,

Where p_{loss} is actual cell loss probability due to the policer and P_{des} the desired cell loss probability when the source violates the negotiated mean rate specified by the network manager. The policing mechanism should tend to shut down sources demanding more resources than negotiated so as to avoid wastage of network resources. Hence the desired cell loss probability P_{des} should be significantly bigger than the ratio $(m-m_o)/m$ which corresponds to proportion of the excess traffic.

If we define the leak rate as $r=cm_o$, where c is a constant equal to or bigger than one, when $r\rightarrow m_o$ that is $c\rightarrow 1$, effective control of the mean rate is achieved. On the other hand, for any value of $r>m_o$ there is a minimum value of Q which assures that the QoS requirements for the cell loss probability are satisfied. This minimum value increases without limit as r tends to $m_o(c\rightarrow 1)$. On the contrary, in order to limit Q, r and consequently c should be increased. In this case, if we define the deviation $d=m/m_o$, then for any value $d<c$, the Lbcan not react. The value of c and consequently the value of r is critical, because it determines how tight the control is. This is very important, because the tighter the control exerted by the procedure, the bigger the bandwidth saved. The maximum allowed burst size BS_{max} is specified by the relation

$$BS_{max}=P*Q/(p-t).$$

Or for the second version

$$BS_{max}=P*B_T/(P-\gamma).$$

The two equations are always valid because always $P>r$ and $P>\gamma$. this is because, as mentioned above, r and γ are usually set equal to or bigger than the mean rate m_o and in the case of bursty source $p>m_o$. It is shown the LB parameter for high speed data as follows:

Table 1 :

Q	C	$Q_{average}$
200	8.013	3.2
2500	1.465	168.7
5000	1.181	443.7
10000	1.077	1047.1
100000	1.005	15693.3

$p=10$ Mb/sec, $m=1$ Mb/sec, $L=100$ cells, $P_{QoS}=10^{-5}$.

Therefore in order to restrict BS_{max}, a small counter and consequently a big r should be used. However this leads to ineffective control of the mean rate. In this case the second version is more effective because it reduces the bursting of the traffic by buffering the incoming cells. For this purpose, several traffic-shaping mechanisms have also been proposed. On the other hand, in order to control the mean rate effectively, a small value of r and consequently, a big Q should be used ; however, this leads t ineffective control of BS_{max}. the average time required to fill the counter immediately after a deviation from the mean rate commences is used as a performance measure of the ability to detect and prevent long bursts. This is given by

$$T_{fill}(d)=((Q-Q_{average})*n_{cell})/((d-c)*m_o), \quad \text{if } d>c,$$

Or

$$=\alpha, \text{ otherwise}$$

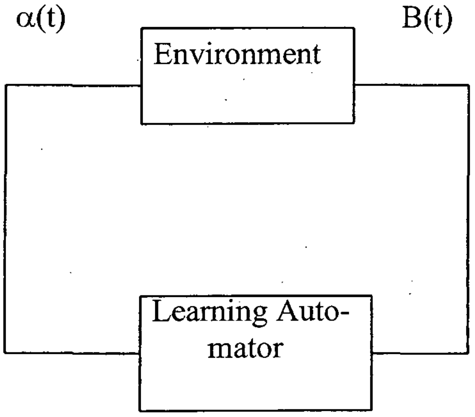
where $Q_{average}$ is the average value of the counter.

The bigger the size of the counter Q, the bigger the reaction time of the LB. Consequently, in order to restrict the reaction time required by the LB to detect a misbehaviour of a source, a small value of Q should be used. However as explained above , in this case the mean rate can not be enforced tightly.

These are the reasons why no matter how the parameters are set, the LB can not control the mean rate and the maximum burst size of the source effectively.

2.5 Learning Automata preliminaries and concepts

A Learning Automaton (LA) is a finite state mechanism that with a stochastic environment given the following figure, and trying to learn the optimal action that the environment offers.



At any iteration, the automation uses an output function triggers the environment that responds with an answer reward or penalty. The automation takes into account this answer and modifies the states to a new one not necessarily different from the former using a transition function.

The greatest potential of the learning automation methology is that it permits the analysis of time complex dynamic systems and global optimisation is possible. Even when little information is available then they tends to stabilise a non-stationary system for predicting its behaviour.

There are various way s of classifying LAs. As far as their Markovian representation is concerned that is proved in the some other reference that an LA is a compact Markov representation process and converges on one of the its, they are classified as ergodic convergence on a specified with a distribution independent of the initial steps. According to the probability values that the actions can take, LAs are characterised or discretised.

A learning algorithm is characterised as an estimator when it uses a running estimate of the probability of each action. The change of the probability of choosing an action is based on its current estimated reward probability rather than on the environmental feed back as in the case of traditional LAs. Finally a learning automaton is called "ε" option, if there is an internal parameter N such as;

$$\underset{N\to\alpha}{\mathrm{Lim}}\ (\underset{t\to\alpha}{\mathrm{lim}}\ E\ \{\ Pm\ (t)\ \})=1$$

(the symbolism E{...} stands for the expectational value).

2.6 Presentation of SELA

The Stochastic Estimator Learning Algorithm used in the proposed mechanism was shown to be a powerful and flexible learning automaton, especially when it operates in a non-stationary stochastic environment. The SELA automaton is an ergodic discretised stochastic estimator one according to the definitions. It is the first attempt in the field of learning automata, as reported which does not utilise the true estimate of the reward probabilities in the search for the optimal action. Instead, it weights the estimates with a Gaussian random variable with a standard deviation proportional to the time that has elapsed from the last time each action was selected. The utilisation of the automaton when it operates in a non-stationary environment. For the reasons of completeness of the h presentation the formal definition is given as that the SELA automation is defined as a sextuple (A,B,Q,P,T,G), where:

- $A=\{a_1,a_2,\ldots\ldots,a_n\}$is the set of the n actions ($2\leq n<\alpha$) offered by the environment. The action selected at time t is symbolised as
  $$a(t)=\varepsilon A$$
- $B=[\ 0,\ 1\ ]$ is the set of the possible environment responses. These can take any value in the [0, 1] space. The environmetal response at a time instant t is symbolised as b(t), while the feedback of the selected action a_k is symbolised as b_k.
- $P=\{P_1,P_2,\ldots\ldots,P_n\}$ is the probability vector that is used to choose an action
- T is the learning algorithm that modifies the probability vector.
- Q is the set of possible internal states of the automaton.
- G is the output function.
- E is defined as an estimator that at any time instant contains the environmental characteristics. Specificially:
  - $E(t)=\{D(t),M(t),U(t)\}$ where
  - $D(t)=\{d_1(t),d_2(t),\ldots,d_n(t)\}$ is the True Estimate Vector,
  - $M(t)=\{m_1(t),m_2(t),\ldots m_n(t)\}$ is the oldness Vector and
  - $U(t)=\{u_1(t),u_2(t),\ldots,u_n(t)\}$ is the Stochastic Estimator Vector.

2.6.1 SELA algorithm

Initialisation: Set all $P_I=1/n$.

Step 1: Select an action $a(t)=a_k$ according to probability vector.

Step 2: Receive the feedback $b(t)\in[\ 0,\ 1\ ]$ from environment.

Step 3: Compute the new true estimate $d_k(t)$ of the mean reward of the action a_k:

$$b_k(t)=\ (\sum_{i=1}^{W} b_k(i))/W\ ;$$

Where $\sum^{W}_{i=1} b_k(i)$ is the total reward received by the automaton during the last W times that action "a_k" was selected . $W\in N$ is a positive integer internal automaton parameter called "learning window" and is used for ignoring old and probably invalid environmental responses.

Step 4: Update the oldness Vector by setting:

$m_k(t)=0$ and

$m_I(t)=m_I(t-1)+1$ for $i\neq k$.

The oldness vector $m_i(t)$ at any instant t contains the time passed expressed in number of iterations from the last time each action was selected.

Step 5: For every action a_l compute the new "stochastic estimate" $u_i(t)$ as:

$$u_i(t)=d_i(t)+N(0,\sigma_i^2(t)),$$

where $N(0, \sigma_i^2(t))$ symbolises a random number selected with a normal probability distribution with mean equal to 0 and standard deviation $\sigma_l=\min\{\sigma_{max},am_l(t)\}$ proportional to the time passed from the last time each action was selected. Specifically,

1. a is an internal automaton's parameter that determines how rapidly the stochastic estimates become independent from the true ones.
2. σ_{max} is the maximum permitted standard deviation of the stochastic estimates that bounds the standard deviation not to increase infinitely.

Step 6: Select the "optimal" action a_m that has the highest stochastic estimate of the mean reward.

Thus, $u_m(t)=\max\{u_i(t)\}$.

Step 7: Update the probability vector in the following way:

For every action a_i ($i=1,2,\ldots,m-1,m+1,\ldots,r$) with $P_i(t)>0$ set:

$$P_l(t+1)=P_l(t)-1/N,$$

Where N is a parameter called "resolution parameter" and determines the step $\Delta(\Delta=1/N)$ of the probability updating.

For the "optimal" action a_m set:

$$P_m(t+1)=1-\sum_{i\neq m} P_lt+1),$$

with the feasibility constrains $0\leq P \leq 1$ and $\sum_{i=1}^{n} P_i=1$.

Step 8: Go to Step 1

A complete formal description of SELA automaton can be found. It should be noted that the choice of the parameters of the automaton (a,σ_{max},N) is critical issue in the relation to its performance under various environments SELA has already been used successfully in the other problems.

Theorem. The SELA learning automaton is ε-optimal in every stochastic environment that offers symmetrically distributed noise. Let $d_1,d_2,\ldots,d_n$ be the mean rewards offered by the environment to the actions $a_1,a_2,\ldots.a_n$ respectively if the action a_m is the optimal one ($d_m=\max_l\{d_i\}$for $i=1,2,\ldots,n$) and $P_m(t)=P[\ a(t)=a_m\ ]$, then for every value $N>=N_0(N_0>0)$ of the resolution parameter there is time instant t_0 < œ such that for every $t>=t_0$ it holds that $E\{P_m(t)\}=1$.

3 **Presentation of the LB-SELA scheme**

3.1. Stochastic fluid model of the Leak Bucket for on-off sources

Using the model of two-state ON-OFF sources which was presented in the above , a very useful fluid-flow approximation was given. According to this approach and using the second version of the Leaky Bucket mechanism, if X and Y denote the contents of the data buffer and token pool respectively, then $W=X-Y+B_T$ is defined as " virtual buffer content" $(0=<W=<B)$.

If $F(\S)=P_r\{W=<\S)$ is the steady-state distribution of W, then

$$F(\S)=(1-\rho\exp(z.\xi))/\Delta(B), \qquad 0\leq\xi\leq B. \text{ -----------\#3}$$

Where,

$$\rho=(p^*T_{OFF})/(\gamma.T_{OFF}+T_{ON}),$$

$$z=(-(T_{ON}+T_{OFF})(1-\rho))/(p-\gamma) \text{ and}$$

$$\Delta(B)=1-((T_{OFF}\,p-\gamma)/(T_{ON}\,\gamma)\exp(z^*B)$$

As mention above, the two versions of the LB coincide if $B_D=0$. So, in order to use the above distribution in the first version of the Leaky Bucket, we should set $\xi=q$, where q = the value of the counter $(0\leq q\leq Q)$. then equation #3 gives the stationary steady-state distribution of the values that the counter can take. This is the one approach used in the proposed scheme.

3.2. Solution of the policing problem using Leaky Bucket with SELA

In the proposed approach, SELA is employed to enhance the performance of the LB. In this presentation, the first version of the LB is used, but , as it will be explained later, the second version can also be used. This new methodology is based on the fact that when one or more source traffic parameters are diverging from their negotiation of the counter is also diverging from the steady-state distribution corresponding to a conforming source. For example in the given the difference between a typical steady-state distribution of the counter when a source transmits at the negotiated mean rate d=1.05 is presented according to the equation #3. Obviously, when the source transmits according to its negotiated mean rate, the observed

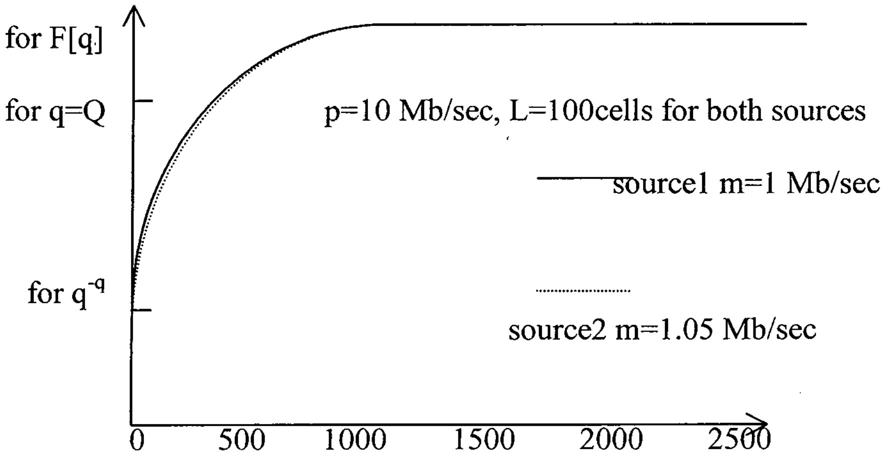


fig.4: Steady-state distribution of the value q of the counter

steady-state counter will converge on the solid curve of the figure. On the other hand when it transmit with $d \geq 1.05$, then the observed steady-state distribution will approximately better the dotted curve of the figure than the solid one. Exploiting this very difference, SELA is trying to detect whether the source is transmitting according to the negotiated traffic parameters or not. Therefore, in order to control the mean rate, two actions are used in SELA that are said to as action 1 and action 2. Action 1 represents the source that transmits with at the negotiated mean rate; and in the action 2, it represents the source that transmits with deviation of the mean rate $d > d_{max}$, where d_{max} is the maximum deviation allowed by the network manager. When the source transmits at the negotiated mean rate, the real steady-state distribution of the value of the counter curve of the figure #4 which corresponds to action 1 ; thus action 1 is rewarded more than the action 2 and SELA converges on the action 1. When the deviation of the transmitted mean rate is equal to or bigger than d_{max}, the real steady-state distribution of the value of the counter approximates better the doted of the given figure #4 which is corresponding to action 2. In this case, action 2 is rewarded more than the action 1 and SELA converges on the action 2.

As mentioned in the above, SELA is trying to determine the behaviour of the source based on the distribution of the values that the counter takes. After extensive simulation runs, it was concluded that for bursty sources which resemble the traffic model of the section the best performance of the LB-SELA is achieved when the feed back $b_I(q)$ of each action I is defined as follows:

$$B_I(q) = F_i(0) \qquad , \text{if } q=0$$

Or

$$F_I(q) - F_i(0) \qquad , \text{if } 0<q<=Q$$

Where "i" is the number of the action 1 or 2.

Using this feedback, it was shown through simulation that a correct convergence of the learning algorithm on the appropriate action is achieved for every source which can be approximated as two states (ON-OFF) bursty source. However employing the appropriate fluid-flow model and redefining the feed back so as to guarantee the correct convergence on the optimal action, the proposed methodology can also be used for the bursty sources which do not resemble the above model as well as for other kinds of the source such as VBR.

It is showing another figure #5 as follow, typically feedback for each of th etwo actions are represented the deviation of the action 2 is set equal to 1.05. the fact that q is defined in a different way for $q=0$ and $q\neq0$ results in a discontinuity in $b_i(0^+)$, which does not have any negative effect on the proposed methodology as it will become evident from the simulation results.

A SELA iteration is executed whenever a burst of cells starts or ends. The value q of the counter at that moment is used to compute the steady-state distribution $F(q)$ for each action using equation #3.

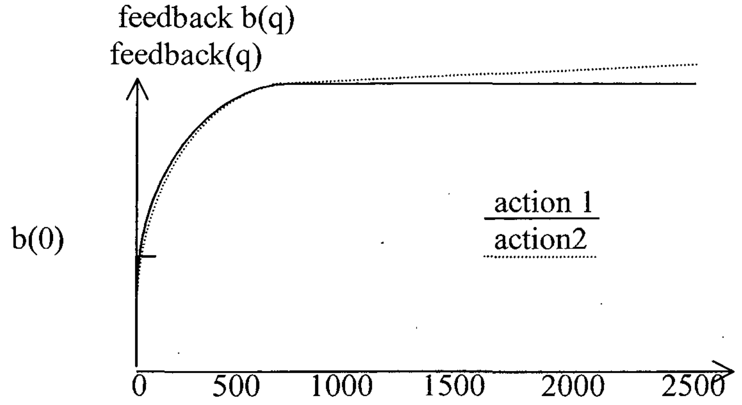


fig.5: Feedback b(t).of SELA for Actin 1 and Action 2

The end of a bursty and an idle period can easily be detected; specially , during a burst duration the value of the Leaky Bucket counter increases with a rate (p-r) until q=Q. On the contrary , during an idle duration

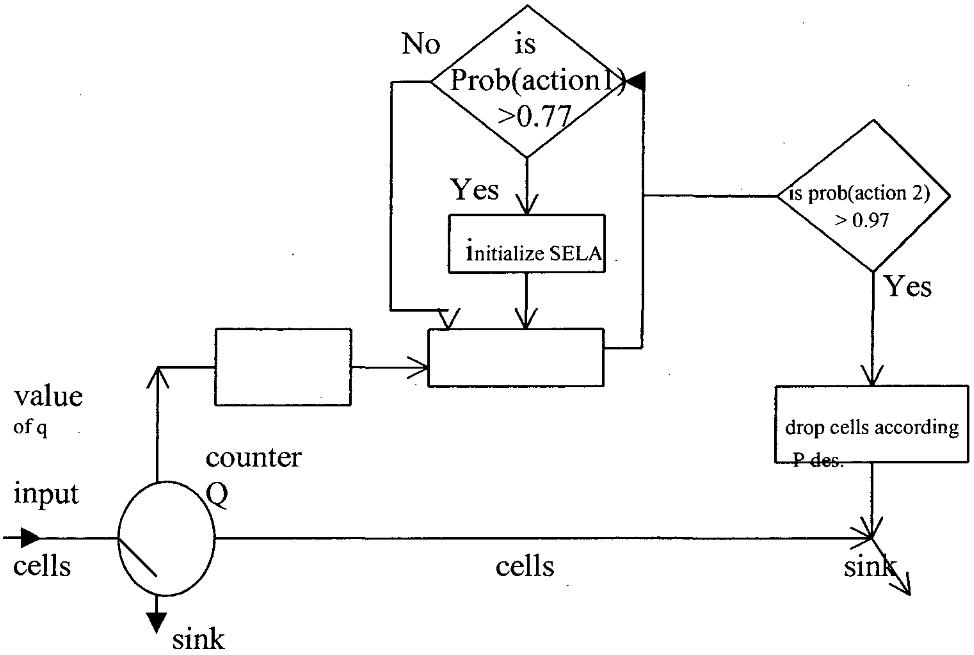


fig.6: LB-SELA

the value of the Leaky Bucket counter decreases with a rate of r until q=0. We avoid running SELA iterations more frequently for example after each cell arrival., because this might lead to understand the desirable delay. The fact that the algorithm samples the counter only at the end of the bursty and an idle period is not a drawback for the control of a long burst. As it will be shown in the simulation results, LB-SELA allows the use of small values of the counter Q and consequently it does not allow the long burst sizes according to the above equations #1 and #2. In the order to use the second version of the LB, equation #3 is used without setting $B_D=0$. In this case $§=X-Y+B_T$ is virtually buffer content. However ever not changes are required for SELA and its actions; in this BS_{max} can be reduced even more. In summary , when SELA converges on the action one it means that the source transmits to its mean rate. If SELA converges on the action two, it means that the source transmits to the deviation equal to or bigger than d_{max}. In this case a threshold T_1 is specified $T_1=0.9$, so that when the prob(action 2)>=T_1, the incoming cells are dropped or marked according to P_{des}. Moreover a source at any instant may change its mean transmission rate. For example while the source transmits with the negotiation mean rate, it may increase it causing congestion to the network. SELA has the ability to adapt rapidly in the changing environments. Consequently LB-SELA as it will be shown in the simulation results, can detect these changes and react suitably and quickly. When SELA converges on action 1, obviously there is no need to take any steps because the source is conforming. However in the order to decrease the reaction time of the LB_SELA in the case of a conforming source which begins to misbehave a threshold T_2 for the prob(action 1) is specified for example 0.7 so that when prob(action 1)>=T_2 SELA is re-initialisation, because SELA needs fewer iterations to converge on the action 2 when for example 0.5<=prob(action 1)<=0.7 than when prob(action1)>0.7. Therefore in the case of the source changing its transmission rate, the measure has a positive effect, whereas in the opposite case ,it does not act negatively. The two thresholds mention above should be determined by the network manager to specify the strickness of the network controls. The influence of the thresholds in the strickness of the network control is a subject to study later on. Because it also be noted that for conforming sources, if the counter reaches the threshold Q before SELA converges on action two, cells are dropped or marked in the same way as at the original LB.

The above fig #6 shows the mechanism of LB-SELA.

Simulation

The goal of the simulation study is to investigate the performance of the proposed methodology for several bursty sources which resemble the traffic model. The confidence intervals of the measurements are 95% constructed with the method of independent replications.

In order to find the optimal values of SELA 's parameters a set of extensive simulation runs are performed, each time changing only one of them while keeping the others constant. Some of the experimental results found from the persons experimental results are given below. These are two experimental results.

Experiment 1: (Table 2)

Negotiated source traffic parameters	LB parameters	traffic parameters of action 1 and action 2		SELA parameter
p_0=10 Mb/sec	Q=2500	$p_1=p_0$	$p_2=p_0$	N=2500
m_0=1 Mb/sec	c=1.465	$m_1=m_2$	m_2=1.05 Mb/sec	$\sigma_{max}=10^{-5}$
L_0=100 cells		$L_1=L_0$	$L_2=L_0$	œ=10^{-6}
T_{On0}=0.00424sec		$T_{ON1}=T_{On0}$	$T_{ON2}=T_{on0}$	W=30
T_{off0}=0.03816 sec		$T_{OFF1}=T_{OFF0}$	T_{OFF2}=0.03614sec	
$P_{QoS}=10^{-5}$				

Experiment 2: (Table 3)

Negotiated source traffic parameters	LB parameters	Traffic parameters action 1 and action 2		SELA parameters
p_0=26.5 kb/sec	Q=100	$p_1=p_0$	$p_2= p_0$	N=2500
m_0=9.3105 Mb/sec	c=1.42	$m_1=m_2$	m_2=9.776 kb/sec	$\sigma_{max}=10^{-5}$
L_0=5 cells		$L_1=L_0$	$L_2=L_0$	œ=10^{-6}
T_{Ono}=0.08sec		$T_{ON1}=T_{Ono}$	$T_{ON1}=T_{Ono}$	W=20
T_{OFFo}=0.147sec		$T_{OFF1}=T_{OFFo}$	T_{OFF1}=0.136sec	
$P_{QoS}=10^{-9}$				

It was amused that the peak rate can be controlled easily and effectively by other Mechanisms as mentioned above. Thus it examine only the effectiveness of the LB-SELA as a function of the mean rate, the mean burst duration and the mean duration, while keeping the peak rate constant. The threshold time T_1 and T_2 are set to be equal 0.9 and 0.7 respectively while P_{des} is set to 0.1.

4.1. Effectiveness of the LB-SELA on policing a source:

Firstly a source with the negotiated traffic parameters of the above experiment 1 is tested. In this case, the appropriate values of the Q and c are given in the Table 1. In the experiments the mean rate m is increased by decreasing T_{OFF} and T_{ON} constant. After extensive simulation runs, it is going to be concluded that for the values in the experiment 1 SELA converges correctly on the action 1 when d=1, and the action 2 when d>=1.05 , that is in the figure 7. In this figure, a dramatic improvement of LB performance is observed. Specifically when d>= 1.465, the LB begins to converge on P_{des}, while the LB-SELA limits the behaviour of the source completely according to P_{des} when d>=1.05. if m is accordingly increasing by increasing T_{on} and keeping T_{OFF} constant, a similar curve is achieved. The figure is shown as follow :

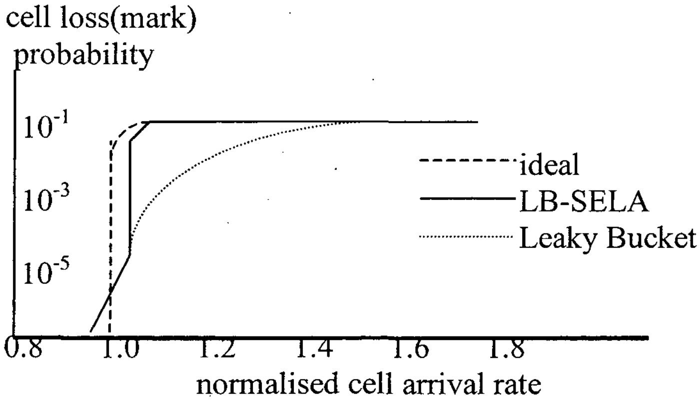
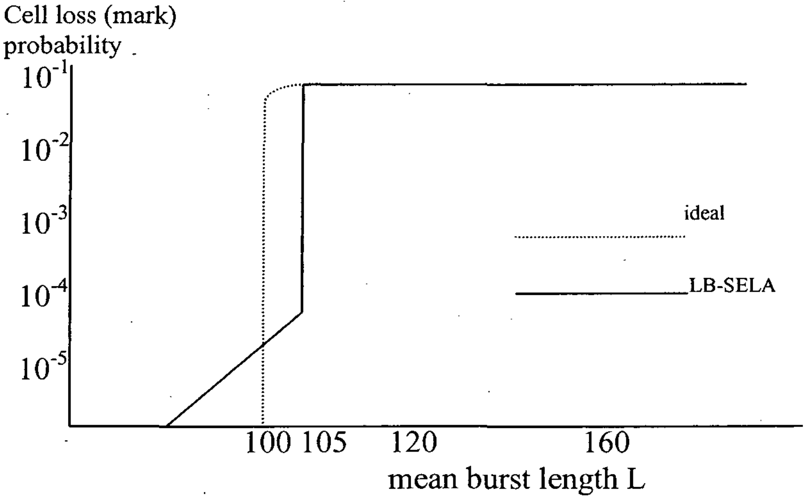


Fig 7: cell loss (mark) probability versus normalised mean arrival rate experiment 1.

In the figure 8, the performance of the LB-SELA as function of L is presented.



fir 8: cell loss(mark) probability versus mean burst length L (experiment 1).

Similar result s for the different sources were realised, as for example packet voice traffic. In the case of a source with the traffic parameters of the table 3, the curve s of the figure 9 (cell lose probability verses mean burst length) are obtained. Obviously, in this case the improvement is also considerable. Thus when d>=1.42 the LB begins to control the source effectively, while the LB-SELA limits it completely when d>=1.05.

It should also be noted that in the case of the sufficiently big deviations from the negotiated values, the real cell loss is bigger than P_{des} because of the cell losses at the counter of the Leaky Bucket. This is not a drawback because, as mentioned in the above section, the goal is to shut down such kinds of sources. Nevertheless , the exact value of P_{des} could be achieved very easily , for example using a counter to count the dropped cells in the first phase in order to drop only the necessary number of the cells in the second phase according to P_{des}. In the simulation runs, the average reaction time $T_{reaction}$ required by SELA to detect the possible non-conforming behaviour of a source is also examined.

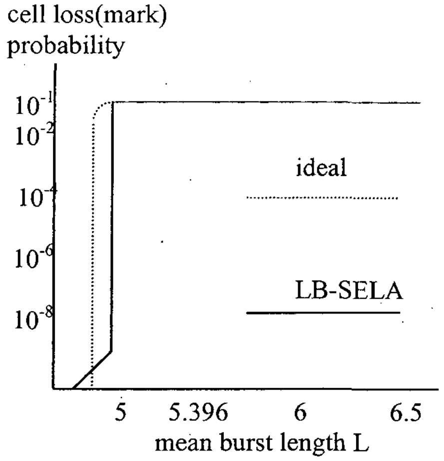


figure 10: cell loss (mark) probability versus mean burst length experiment 2

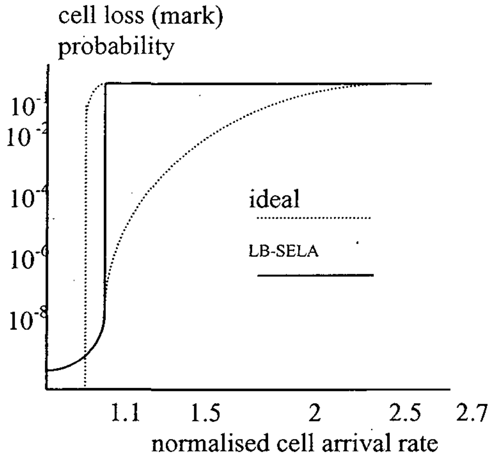


figure 9: cell lose (mark) probability versus normal mean arrival rate experiment 2.

It should be mentioned that the reaction time does not have the same importance as T_{fill} as described in the above. $T_{reaction}$ expresses the average time required by the LB-SELA to detect a misbehaviour and limit the source completely according to P_{des}, while T_{fill} is the average time required to fill the counter as soon as a deviation from the mean rate commences. Nevertheless, in the case of small deviations from the negotiated values, that is for difficult " control areas" $T_{reaction}$ is significantly smaller than T_{fill}. In the following figures $T_{reaction}$ is presented. For example, in the figure 10, when $T_{reaction}$=26.989 sec. It should be noted that the initial state of the LB-SELA parameters in these experiments is q=0 and prob(action 1) = prob(acton 2) = 0.5.

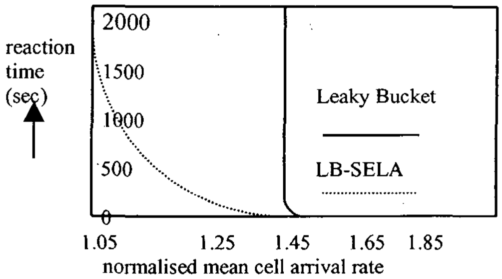


figure 11: reaction time versus normal cell arrival rate experiment 1.

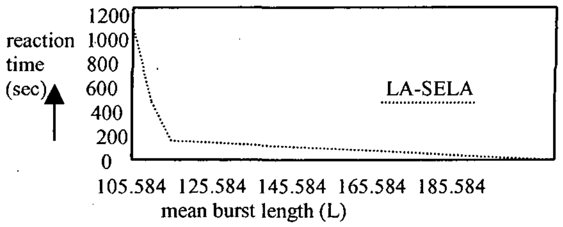


figure 12: reaction time versus mean burst length L experiment 1.

As explained in the above, when the leak rate r decreases, the size of the counter Q should be increased in order to guarantee the QoS constraints, and the contrary. Moreover it was shown that the Leaky Bucket requires the smaller leak raate than the LB-SELA approximately equal to the negotiated value so as to control the mean rate effectively and thus it also requires smaller a bigger counter. Therefore, using equation#1.1, it is concluded that the LB-SELA enforces the maximum allowed burst size BS_{max} which can enter the network better than the LB. For example in the experiment 1, the LB needs c=1.077 in order to control deviations d>=1.077 and so, a counter with size Q=10000 is required according t the table 1. In this case, according to equation#1.1 the maximum burst size that is allowed to enter the network is BS_{max}. on the other hand, LB-SELA needs r=1.465, Q=2500 and thus BS_{max}. Thus achieves tighter enforcement as far as BS_{max} is concerned than the LB.

The performance of the proposed methodology in the case of changing rates is examined by the extensive simulation runs. When a conforming source at he certain times starts to transmit with a mean rate bigger than the negotiated value, the LB-SELA reacts accordingly. For example it will be shown in the figure 15, in that case a conforming source starts to misbehave with the value d=1.4, the reaction time of the LB-SELA depends on the value of the prob(action 1) at the very moment that the change happens. As it can be observed in the worst case when the prob(aactoin 1) = 0.63, $T_{reaction}$=39.338 sec, which is just a little bigger than the results in the figure 11 remembering that when d=1.465 for the LB $T_{fill}=\infty$. The LB-SELA stops dropping incoming cells when prob(action2) < 0.9. Even tighter control can be achieved with down limit equal to prob(action 1) = 0.5.

Thus in the order to police a burst source according too P_{QoS}, Q, c and the appropriate values of SELA's parameters which vary a little from case to case are chosen. It should be noted that in the experiment 1 and the tables 2 and 3 , two sources with extreme values of traffic parameters in the possible set of values that they can take are used. When the traffic parameters for that source lie between the values f SELA's parameters used in the experiments.

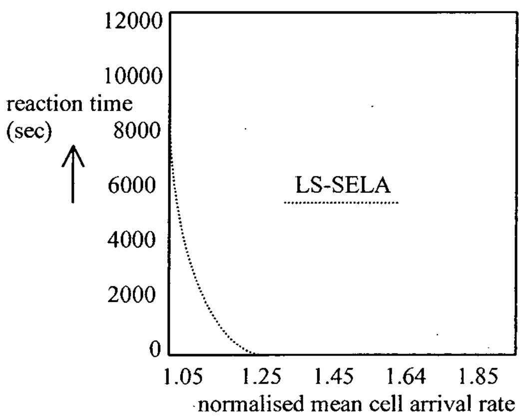


figure 13: Reaction time versus normalised mean cell rate experiment 2

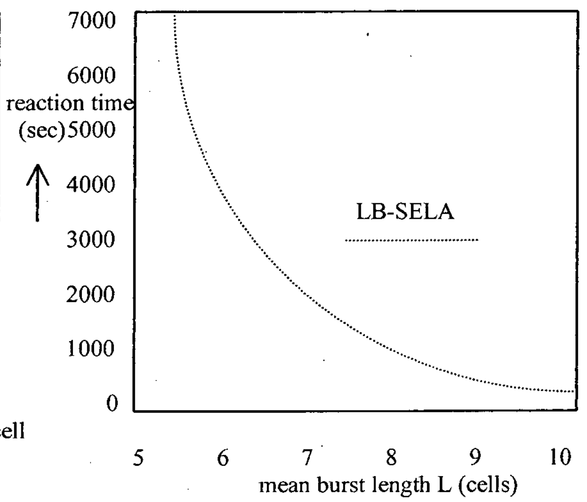


figure 14 : Reaction time versus mean burst length L experiment 2.

4.2 Effectiveness of the LB-SELA in an inter-nodal node:

From the many papers and magazines in the literature up to now have examined the effect of the several traffic enforcement mechanisms on the source which they police. However the non-conforming sources may affect the conforming ones which share the same buffer in the inter nodal nodes. Hence the most important issue regards the examination of the influence of such sources on an inter-nodal link queue when they are policed by a traffic enforcement mechanism. It is shown that traffic generated by non-conforming sources can pass through the LB without being discarded and cause unacceptable cell loss and delay performance at an inter nodal node. Here, in this section the effect of LB on such a node when one or more sources diverges from their negotiated mean rate is also examined through simulation.

In the experiment the network model of the figure 15 is used.

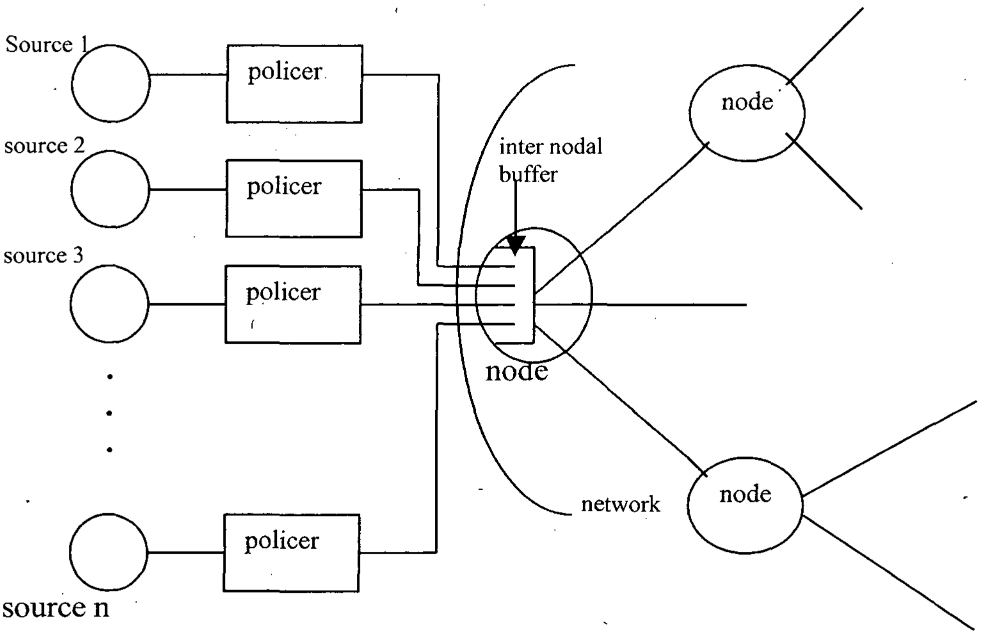


figure 15 : Network model

In the figure firstly the effect of the three sources with the negotiated traffic parameters and the corresponding SELA's parameters described in the table 2 are examined. It is assumed that the average session length is 100 sec, the size of the buffer at the node is B=100 cells and the link capacity is C=20 Mb/sec. Obviously, in the real ATM network the link capacity is bigger, but there are also more than three sources. The results when one of the 3 sources diverges from its negotiated mean rate with deviation d=1.4 are presented. It can be noticed that the lost cells of the conforming sources in the buffer of the node are fewer when the LB-SELA is used. Specially the average lost cells of the conforming sources are only 23.889 when the LB-SELA is used 123.709 when the LB is used. That is because SELA drops about 227492 cells of the non-conforming source and the counter of the LB drop only 4512.8 cells. Thus the total dropped cells by the LB-SELA are 232004 and by LB only 4512.8. At this point it should be noted that the total dropped cells by the LB-SELA are the sum of the cells which are dropped by the SELA and dropped by the counter. On the other hand, the total cells by the LB are only the cells which are dropped at its counter.

The results when two of the three sources diverge from their negotiated mean rate with d=1.4 are presented. The difference in the performance between the two methodologies is even greater. The average lost cells of the conforming source in the buffer of the node are 25.809 when the LB-SELA is used, and when the LB is used. It should be noted that the counter of the LB drops some cells of the conforming source, but SELA does not drop any cell. This means that the SELA can understand much better than the LB whether a source is non-conforming one and further more to shut down it or whether it is a conforming one. The corresponding numerical results are presented when all the sources transmit according to their negotiated values. It can be noticed that the average lost cells in the buffer of the inter-nodal node are 53.419, which are more than the average lost cells when one or more sources misbehave and the LB-SELA is used. That is so, because the LB-SELA shuts down the non-conforming ones. However, this is not the case with the LB, as it is derived from the results of the three tables above.

The results are presented when the size of the buffer at the nod is B=a50 cells and two of the three sources diverge from the negotiated mean rate with d=1.4. It can easily be noticed that the average lost cells in the both cases are fewer than the average lost cells that are presented due to the buffer size, but there are significant different between LB-SELA. When the LB-SELA is used, the average lost cells of the conforming source at the node are only 5.52 and when the LB is used, they are about 44.9 which are more than in the case where B=100 and the LB-SELA is used. Consequently when the LB-SELA is used as a traffic enforcement mechanism, smaller buffers at the nodes can be used resulting in the few cells losses than the LB with the bigger buffer is used. It should be noticed that the cells loss percentage at the node is $5.552/238153=2.31*10^{-5}$ when the LB-SELA is used , and $44.9/238153=1.88*10^{-4}$ when the LB is used, a difference of almost at the order of magnitude.

Let us examine a source which is a transmission with the deviation of the mean rate d=1.7. After it is examined, the LB manages to control sources with big deviation of their negotiation values very effectively. However, even in this case it can be noticed that, when the LB-SELA is used, the average lost cells of the conforming source in the inter-nodal buffer are about 9.4 and when the LB is used, they are about 125.37. That is because, when the LB-SELA detects a mis-behaviour, it limits it completely, as mentioned earlier. Thus the cell loss percentage is $9.4/237428=3.95*10^{-5}$ when the LB-SELA is used, and $125.37/237428=5.280*10^{-4}$ when the LB is used, a different of more than an order of the magnitude. The average lost cells when there are more than three sources were also examined. The results when three of the five sources transmit with deviation of the mean rate d=1.4 and d=1.7 respectively are presented. As it is shown, the results are similar to those in the previous experiments. LB-SELA is able to control the non-conforming ones and to limit the cell losses at the node.

Similar results for various types of the bursty sources were also obtained. For example, the results are presented correspondingly to sources with the negotiated traffic parameters given at the table 3. It can easily be noted that the performance of the LB-SELA is better than the one of the LB. It should also be noted that the fact that the LB-SELA limits the excess traffic of the conforming source better than the LB leads to the smaller queues at the buffers of the inter-nodal nodes and thus to smaller cell delays. The average queue size at the buffer is in the table 4 below . in the table 4 , the average queue size at the buffer is about 71.404 cells and consequently when the LB-SELA is used, the average cell delay is about 0.865 sec. On the contrary, when the LB is used, the average queue and the average delay are about 179.375 cells and 2.173 sec respectively. It is examined the effect of the mean rate only in this section, because this is the most critical traffic parameter. However as shown in the above, LB-SELA is able to control effectively the source parameters too. The results are similar when any other source traffic parameters diverge from their negotiated values.

Table 4:

Effect of the LB-SELA in an inter-nodal node with three total sources, one non-Conforming source with deviation of the mean rate d=1.5.

	Average queue at the buffer Of the node (cell)	Average delay introduced by the buffer of the node (sec)
LB-SELA Is used	71.404 ±1.29	0.865±1.01
LB is used	179.37±1.02	2.173±1.01

In summary , it is shown by the simulation that the proposed methodology achieves faster, tighter and more effective control of the bursty sources which can be approximated by the two states Markov model. The statistical gain obtained using the LB-SELA leads to fewer cell losses and to smaller cell delays at the buffers of the inter-nodal nodes and so, it guarantees the QoS constraints much better than the original LB.

5. Conclusions and future work:

In the above sections, an effective policing mechanism is presented, which order to determine whether a source violates its negotiated traffic parameters. For this purpose, the distribution of the values of the Leaky Bucket counter is employed. As the numerical results demonstrate, the proposed scheme is able to achieve faster and tighter control of a source, and thus, more statistical gain and better guarantee of the QoS constraints. The more effective policing of the source by the purposed methodology results also in the fewer cell loses and smaller delays in an inter-nodal node where many sources share the same buffer. Besides this, the new scheme, although it requires more computations than the original LB, it does not involve excessive information storage or computational overhead, while the required updating time lies within an acceptable range for real time applications. Moreover, learning algorithms are amenable to simple hardware implementation, as basic stochastic computing elements enhances the Leaky Bucket with a powerful learning algorithm, called SELA, have the proved their synthesising learning automata.

As mentioned above, the proposed methodology policing only bursty sources which can be approximated by the two state Makov model in order to employ this mechanism to enforce other kind of sources such as VBR sources, the appropriate fluid-flow models should be used. However, in these cases the feedback that the LB-SELA receives should be defined in order to achieve a correct convergence of the learning algorithm.

In the forthcoming, the LB-SELA for VBR traffic sources must be presented for the future objective. The future objective is the hardware implementation of the LB-SELA and the exact definition of SELA's parameters for every case. Finally, for this section as a target objective, it is necessary to learn algorithms and performed by a single chip multi-plexer for ATM networks.

CHAPTER 3: Analysis of rate bit control in the ATM network

1. Introduction :

With the advancements in the computer technology, computer communication protocols are getting larger to fulfill more complicated tasks. However, this complexity means that they may also be less reliable. Consequently, validation, verification, and testing become an dispensable part of the protocol design and implementation; yet it has proved to be a formidable task for complex systems. Only very few protocol systems have been analyzed formally. A major hurdle is that formal analysis often requires a formal specification of the protocols, yet a large number of the protocols in the practice are specified informally, mostly in english.This section provides an analysis of the source policy in the rate-based congestion control scheme developed by the asynchronous transfer mode (ATM) forum for the available bit rate service and derives approximate analytical closed-form expression to describe the rate increase process. These approximations are used to analysed the impact of the source algorithm on the TCP slaw-start process operating over a rate-controlled ATM network. The results show the increase in the TCP congestion window came-up time is noticeable when Th round-trip delay is small. The results are verified by the simulation. Broadband packet networks based on the asynchronous transfer mode (ATM) are enable the integration of the traffic with a wide range of characteristics within a single communication network. Many service classes have been defined in the ATM for the support of the traffic with different quality of the service (QoS) requirements. The available bit-rate(ABR) service class is defined for the best effort applications that may require a minimum-bandwidth guarantee. This service class allows applications to fully utilized the available network bandwidth without causing congestion and to provide a fair allocation of the available bandwidth among the connecting shares of the network. The ATM forum traffic management committee recently completed the specification of the rate based congestion control framework to satisfy this objection. In the rate based congestion control framework development by the ATM forum, the source of each connection periodically transmits especial resource management (RM) cell that is used to monitor the congestion state of the network and signal it back to the source. The congestion control framework defines two components :

- The behavior of the source and destination end systems and
- The behavior of the network elements (switches).

The congestion control function within the switches is responsible for the signaling their congestion state information by the adjusting the rate of the transmission using an increase or decrease policy.

In this chapter, it is interest to analyze the rate increase process at the source during startup. In the ATM Forum scheme, the source initially starts its transmission at an initial rate, called the initial cell rate (ICR), chosen based on the factors such as the round trip delay of the connection and buffer sizes in the individual switches. As RM cells are received back from the network, the rate increase algorithm is based on the additive increase, that is, increasing the transmission rate by a constant amount on the receive of the RM cell. Assuming stable network conditions, the time to reach steady state is determine by the parameters of the source algorithm, as well as the round trip delay of the condition. Our basic objective in this paper is to develop simple approximate analytical models to determine the time taken by the source to reach the maximum rate allowed by the network. Such a model would enable a more intelligent selection of the parameters of the source algorithm. A second objective of the modeling ATM source behavior is to evaluate its effect on the behavior of the higher layer protocols. For example TCP incorporates the slow start algorithm to increase its window progression at start up, or upon packet losses.

As a case study of deriving a formal specification from an informal description, it study the ATM Forum Traffic Management Working Group's specification for the ABR service. The ABR service is meant to address the needs of the bursty data transfers, and therefore is based on the effective operation of the end- end rate based congestion control or avoidance mechanism. Sources adjust their rate such that the aggregate load on the network does not exceed the capacity of the network, including an option called Explicit Forward Congestion Indication (EFCI) scheme and a different option called the Explicit Rate (ER) scheme. The specification for the ABR service is primarily that of the source and destination policies, while the specification of the operation of the intermediate nodes(switches) is somewhat more loose to allow for vendor latitude. In the broad terms, the ABR scheme has sources transmitting data cells at a specified rate, derived using feedback control mechanism by which the network communication is the its own. Periodically, a source transmits a resource management cell(RM cell). This RM cell serves the function of a probe into the network to detect the state of the network congestion or otherwise. The RM cell serves as a cell in which the network may communicate an explicit rate back to the source is permitted to transmit. RM cells sent by a source are received by the remote destination and have to be turned around and sent back. This function is performed by the destination of the ABR specification. The source and destination policies have been specified using an English description in the main body of the traffic management specification. Given a protocol of such complexity, it is out of The question to perform a formal analysis manually, such as with the assertion proofs. It is also impossible to conduct an automatic verification, validation, or testing since computer aided techniques require a formal specification and there are no general procedures to obtain such a specification from the english descriptions. Furthermore, while considering energy has been spent in providing in a reasonable precise specification, an english description may sometimes lead to implementation that do not meet the letter and/or the spirit of the specification. Communicating and extending the final finite state machines have been used to model protocols. To properly model the parameters in the ABR protocol associated with input or output data of the RM cells, it need another level of the generation : it use a parameterized extended finite state machine with times (PEFSM) to model and specify the ABR source or destination behavior. As in a finite state machine, there are states and transitions between states. In addition, it has a predicate or event and an action; a transition is executive if the associate predicate is TRUE with the current variable values. If the predicate is TRUE, the action updates the variable values and the machine moves to the next state. Furthermore, there are parameters associated with the input or output cells to or from the machines and they effect the execution of the transition and the variable values. In addition, there are times and the machine behavior also depends on the values for these timers. Such a representation is also called a transition system. implied in the english specification in the

ATM-ABR protocol is the need for a scheduler at the source which pace out data and RM cells. Data cells are transmitted at a rate of ACR , the currently Allowed Cell Rate. After sending a certain number of the data cells, a forward RM (FRM) cell is transmitted by the source, maintaining the same ACR rate. FRM cells received from a remote station are turned around and transmit back as the turned around Backward RM (BRM) cells. Furthermore a fairly stringent rule has been specified for the ordering of the transmission of the different types of cells from the source: data cells vs. FRM cells vs. turned around BRM cells. To pace out these difference types of the cells, a schedule is needed. It identify the minimal requirement for a scheduler to match the specification. The protocol specification requires the source to transmit the FRM cells based on a timer when the normal time between the successive FRM cells is greater than the threshold. Also associated with such a timer based on a transmission is a rate change, specified in the protocol. All of these require a precise interface between the source and scheduler machines.

In summary, for the formal specification we have three machine: source, destination, and scheduler. For each machine, we have a set of states, variables, and a list of transitions between states. It identify the minimal requirements for the scheduler and the minimal interface between each pair of the three machines to match the specification accurately. This chapter is as follows that in the next section provides an overview of the organization and interdependence between the state machines. It presents the formal specification of the source, destination and scheduler machines in the three sections and in each section the transition of the PEFSM is presented associating and comparing them with the rules in the original specification.

2. Organization of the state machines:

The source and destination protocols for the ABR traffic management scheme are part of a closed loop feed back control system, and furthermore the virtual circuit(VC) is full-duplex. The source and destination end system of the VC have the end-end protocol running in them and involve both sending information as well as receiving feed-back from the remote system. it is shown in the figure 1 of this chapter below:

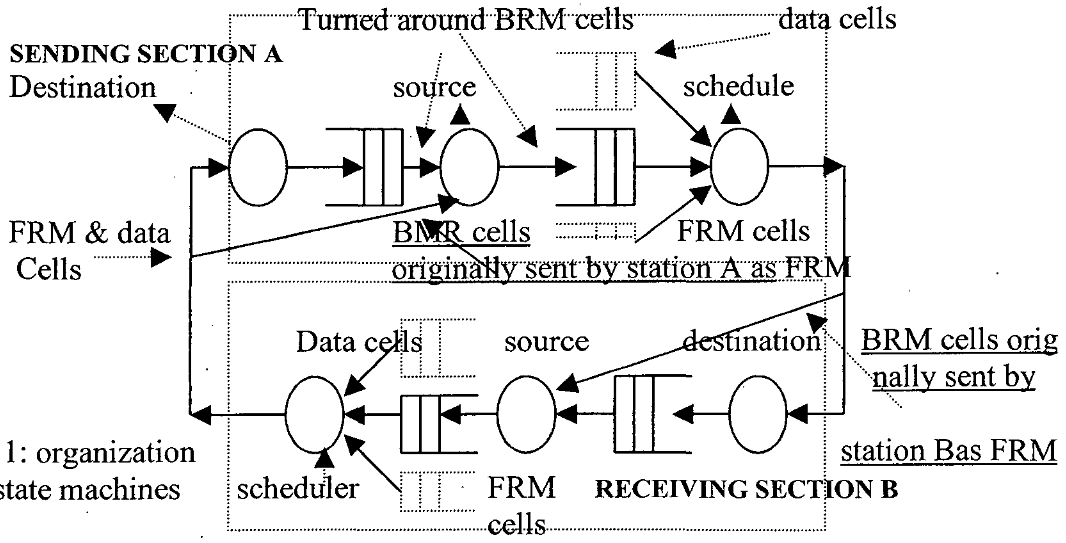


Figure 1: organization of the state machines

In the figure , the relation between each pair of the state machine at both the sending station A and receiving station B are considered. The sending station A, has a source protocol machine which is involved in the initialization, sending data cells to station B via a scheduler machine. It break up the source behavior into 2 parts. A source machine that models all of the non-timing related functions of changing the rate, and servicing BRM cells queued by the destination for the opposite direction. These BRM cells are then queue to the second machine. The second machine is called the scheduler which performs all the timing related functions of the source specification. Specifically source rule 3 and 5 are incorporated in the scheduler machine. The interface between the source protocol machine and the scheduler machine, say station A, has the following three queues:

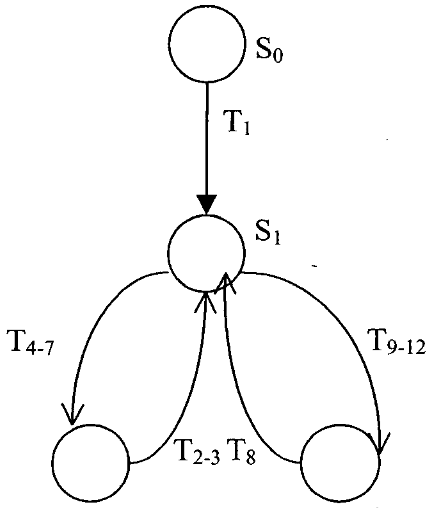
1. A queue of the turned around BRM cells that have to be sent back to station B. This queue is between the source and the scheduler.
2. Two virtual queues of the data cells and FRM cells. When the scheduler machine determines it is time to send an FRM cell , for example station A generates an FRM cell every Nrm-1 data cells, it creates an FRM cell (as if picking up an FRM waiting in its virtual queue) fills in the relevant information in its fields and transmits it.

Another virtual queue for the data cells , is used by the scheduler machine to pick up a data cell from the user to transmit when it is time to send a data cell. The scheduler Machine keeps tracks of time T, which is the since the transmission of the last FRM cell, and also the time t between Th successive transmission of cells of the VC, which are separated by the time interval 1/ACR, specified by the source or destination policies. ACR is the current allowed cell rate at the source. An FRM cell received at a station for example station B comes into the destination protocol state machine, and is converted into a turned around BRM cell. This turned around BRM cell has to be transmitted back to the other station that is station A in the figure 1 of this chapter. It goes through an interface between the destination protocol machine and the source protocol machine at the station B, through a queue that only contains turned around BRM cells. Then the source machine hands the turned around BRM cell to the scheduler state machine at the station B which transmits it on the link. The cell is now delivered s a BRM cell to the source protocol machine at the station A. the source protocol machine receives this BRM cell and takes the appropriate action typically by the alternative the source transmission rate ACR. We therefore have the destination protocol state machine interface with the source through a queue for turned around BRM cells the source through a queue for the turned around BRM cells. The source protocol state machine through three queues for the data, turned around BRM cells and FRM cells respectively. The scheduler finally transmits the appropriate cells based on the protocols specification. It describes in the following sections the three machines : the source, the destination, and the scheduler. Each section begins with a description of the machine followed by its formal specification using PEFSM.

3. Source machine (source behavior):

The main pair of the source behavior modeled here are the actions that change the rate upon the arrival of a BRM cell return from the remote destination, and the interface between the destination protocol machine and the scheduler to transmit turned around BRM cells. The interface with the destination protocol machine is through a queue of the BRM cells that it queue with the source machine. It also provides some of the properties of the PEFSM relating to the source machine. The initial transition $T_1$is occurred when the connection is set up of source rule 2. The first action perform is to initialize the rate to ICR and move to an action state. When a BRM cell arrives, transition T_2 or T_3 is tried leading to one of the transitions T_4 through T_7 where a rate change modification takes place.

Initially the interface between the source and the destination machine is typical of many bi-directional protocols when a protocol machine receives information and has to turn around and transmit some corresponding information for example acknowledgement. In the case of the ARB protocol, the destination protocol machine turn around FRM cells received as turned-around BRM cells. These have to be merged into the stream of the cells sent in the forward direction along with the data and the FRM cells. We seek a minimal interface between the destination protocol machine and the source protocol machine. Instead of keeping track of the individual RM cells queued by the destination protocol machine to the source machine, we maintain a simple counter of the number of the turned around BRM cells queued to the source by the destination machine and a single copy of the contents of the RM cell queued. This is because it is able to exploit the nature of the interface, which is that all the turned around BRM cells queue at the source are either dropped or updated with the latest RM cell information received by the destination. This allows us to retain the semantics of the all the machines without having to keep a queue of all of the individual RM cells queued and having to search through the queue when altering these RM cell, as required by the destination rules, described in the following. When a BRM cell is received by the source, this is the return of the feedback information from the network. This feedback information is then used by the source to modify the transaction rate, ACR. Transmission T_2 and T_3 recognize the receipt of the BRM cell from the remote end, and capture the information returned in the RM cell when the source gets feedback from the switches and possibly by the destination operating in the EFCI mode.



We distinguish by a bit, the BN_{BRM} bit, two types of the BRM cells; it assign BN_{BRM} to 0 for those BRM cells received that have made the entire round-trip; and assign BN_{BRM} to 1 to those BRM cells that are generated from the network in a Backward Explicit Congestion Notification mode. Transitions T_4 and T_7 reflect the manner in which the source modifies the ACR, when it is responding to switches operating in both the EFCI mode as well as the ER mode. The CI field in the returned RM cell takes effect to reduce the rate if any, in transition T_4, and then explicit rate, ER_{BRM} , returned in the BRM cell is used to reduce the source's ACR subsequently. Similarly, increases to ACR occur only if the switches or destination operating in the EFCI mode do not indicate via a feedback that they are congested, and the explicit rate returned from ER (Explicit Rate) switches allows ACR to be increases from its current value. The transition of the source machine is shown in the figure 2 above.

4. Source machine:

4.1. States

S_0: initial connection set up and first data cell available
S_1: active state
S_2: state in which rate may be changed
S_3: state in which queue turned around BRM cells may be altered

4.2 Variables:

N: number of turned around BRM cells in queue from destination to source (share variable with destination)
X: number of the turned around BRM cells in queue from source to scheduler (share variable with scheduler)
Y: number of FRM cells sent by scheduler after initialization and before receiving the first BRM cell with BN=0 or since last BRM cell with $BN_{BRM}=0$ receive by the source (share variable with scheduler)

4.3 Events

E_1: BRM cell received by the source
E_2: data cell waiting to be sent

4.4 Transitions

T_1 (Source 2)
Current state: S_0
Next state : S_1
Event: connection set up complete
Actions: ACR:<= ICR (ACR set to a value less then or equal to ICR)
N:=0;
X:=0;
Y:=0;

```
$T_2$(source 8)
    current state: $S_1$
    next state: $S_2$
    event:
        $E_1(BN_{BRM}=0)$
    actions:
        $CI:=CI_{BRM}$;
        $NI:=NI_{BRM}$;
        Y:=0; (lock Y)
$T_3$(source 8)
    current state: $S_1$
    next state: $S_2$
    event:
        $E_1$ $(BN_{BRM}=1)$ (destination generated BRM cell)
    actions:
        $CI:=CI_{BRM}$
        $NI:=NI_{BRM}$
$T_4$(source 8 and 9)
    current state: $S_2$
    next state: $S_1$
    event:
        CI=1
    actions:
        $ACR: <= \min\{ACR*(1-RDF),ER_{BRM}\}$;
        $ACR := \max\{MCR,ACR\}$;
$T_5$(Source 8 and 9)
    current state: $S_2$
    next state: $S_1$
    event:
        $((CI=0)$ & & $(NI=0))$ & & $(ACR<ER_{BRM})$
    actions:
        $ACR:<= \min\{ER_{BRM,} PC=ACR+RIF*PCR\}$;
        $ACR := \max\{MCR,ACR\}$;
$T_6$(Source 8 and 9)
    current state: $S_2$
    next state: $S_1$
    event: ((CI=) & & $(ACR<ER_{BRM}$ (NI=1 prrevents a rate increase)
    actions: (no increase in ACR)
$T_7$(Source 8 and 9)
    current state: $S_2$
    next state: $S_1$
    event:
        $(CI=0)$ & & $(ACR>=ER_{BRM})$
    actions:
        $ACR: <=ER_{BRM}$ ;( ACR set to value upto returned ER)
        $ACR:= \max\{MCR,ACR\}$;
```

T_8(related to destination 3)
current state: S_1
next state: S_3
event: N>0
action:

T_9(destination 3; source queue turned around BRM cell)
current state: S_3
next state: S_1
event: (X=0) & & (BN_{BRM}=0) (scheduler queue has no BRM cells)
actions: N:=N-1; (lock N)
queue this turned around BRM cell to scheduler;
X:=1; (lock X)

T_{10}(destination 3; source queues turned around BRM cell for BRM dropping)
current state: S_3
next state: S_1
event: (X>0) & & (BN_{BRM}=0) (scheduler queue has BRM dropping
actions: N:=N-1; (lock N)
drop all queue turned around BRM cell at scheduler;
queue this turned around BRM cell to scheduler;
X:=1; (lock X)

T_{11}(destination 3; source queue turned around BRM cell for BRM rewriting)
current state: S_3
next state: S_1
event: (X>0) & & (BN_{BRM}=0) (Scheduler queue has BRM cells)
actions: N:=N-1; (lock N)
rewrite all queued BRM cells at scheduler queue;
queue this BRM cell to scheduler;
X:=X+1; (lock X)
(for an implementation, either T_{10} and $T_{11}$is used for BRM dropping and rewriting respective; but not both)

T_{12} (destination 5; source queue destination generated BRM cell with BN_{BRM}=1)
current state: S_3 (by destination 5, no BRM ells altered)
next state: S_1
event: (BN_{BRM}=1)
action:
N:=N-1; (lock N)
assign values to the various fields of the BRM cell;
queue this BRM cell to scheduler; (destination generation generated BRM cell with BN_{BRM}=1 does not alter the queued BRM cells in scheduler)
X:=X-1; (lock X)

5. Destination machine (destination behavior):

The destination protocol machine is somewhat simpler than thesourceandscheduler machines, even though there are almost the same numberof the transitions. There are several transitons in the destination machine used used primarily to reflect all variants of the network operating in the EFCI and ER modes. This affects the destination transitions and the corresponding actions when it gets an FRM cell and has to prepare it to send it back as aturned-aropnd BRM cell.

The primary event that the destination machine operates from is the arrivalof an FRM cell at the destination. This is causing the destination to turn around the cell as a BRM cell according to the destination rule 2 of the english specification. We reflect this in transitions T_2 through T_8. The alternatives of whether to set the No Increase(NI) bit is implemention specific. The events for the corresponding transtions may not be mutually execlusive, resulting in non-determinism in the EFSM model. In each case, the appropriate fields of the BRM cells are set as specification in the corresponding english rule, and we move to a state where the BRM cella are queued to the source.

Transition T_{11} finally hands the BRM cell to the source machine, for it to then 'transmit' tha is hand to the scheduler. The interface between the destination machine and the source machine is minimal one. The only shared variable is the counter N, which is ncremented in transition T_{11}, when a turned-around BRM cell is queued.

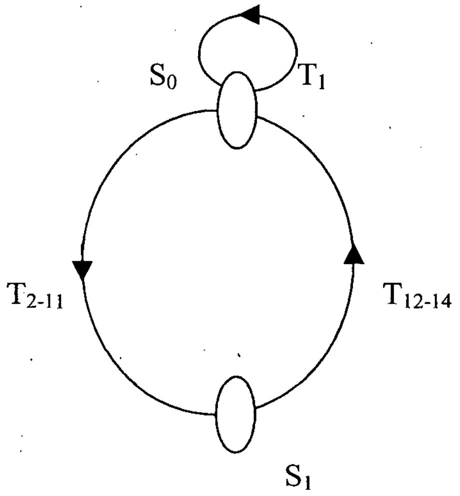


Figure: Destination machine

The transition diagram of the destination machine is shown above.

6. Destination machine:

6.1. States:

S_0: wait for cells

S_1: state in which turned around BRM cells are queued to source and source queue of turned around BRM cells may be altered.

6.2 Variables:

save_EFCI: EFCI state of connection saved at the destination

N: counter of the number of turned-around BRM cells in queue from destination to source (shard variable with source)

6.3 Events:

E_1: an FRM cell received
E_2: a data cell received
E_3: destination has internal congestion

6.4 Transitions:

T1 (destination 1)
current state: S_0
next state: S_0
event:
E_2 ($EFCI_{DATA}$)
actions: saved_EFCI := $EFCI_{DATA}$;

T_2 (destination 2)
Current state: S_0
Next state: S_1
Event: E_1 & (!E_3) & & (saved_EFCI = 0)
Actions: DIR:= 1;
BN :=0;

T_3 (destination 2a)
Current state: S_0
Next state: S_1
Event: E_1 & & (!E_3) & & (saved_EFCI=1)
Actions: DIR :=1;
BN :=0;
CI := 1;
Saved_EFCI :=0;

T_4 (destination 2b, for ER destination)
Current state: S_0
Next state: S_1
Event: E_1 & & E_3 & & (saved_EFCI =0)
Actions: DIR :=1;
BN :=0;
Saved_EFCI :=0;
Reduce ER if desired;

T_5 (destination 2b, for ER destination)
Current state: S_0
Next state: S_1
Event: E_1 & & E_3 & & (saved_EFCI =1)
Actions: DIR:=1;
BN:=0;
CI:=1;
Saved_EFCI :=0;
Reduced ER if desired;

T_6 (destination 2b, for the non-ER destination)
Current state: S_0
Next state: S_1
Event: E_1 & & E_3 & & (saved_EFCI =0)

Actions: DIR:=1;
BN:=0;
NI:=1;
Saved_EFCI:=0;

T_7 (destination 2b, for the non-ER destination)
Current state: S_0
Next state: S_1
Events: E_1 & & E_3 & & (saved_EFCI = 0)
Actions: DIR:=1;
BN:=0;
CI:=1; (T_6 and T_7 exhibit a non deterministic behavior of the system; choose one for implementation; it choose T_6 for analysis)

T_8 (destination 2b, for non-ER destination)
Current state: S_0
Next state: S_1
Events: E_1 & & E_3 & & (saved_EFCI=1)
Actions: DIR :=1;
BN:=0;
CI:=1;
Saved_EFCI:=0;

T_9 (destination 5, cause rate reduction)
Current state: S_0
Next state: S_1
Event: E_1
Actions:
DIR:=1;
BN:=1;
CI:=1;

T_{10} (destination 5, prevent rate increase)
Current state: S_0
Next state:S_1
Event: E_3
Actions: (destination generates BRM cell)
DIR:=1;
BN:=1;
CI:=0;
NI:=1;

T_{11}(relating to destination 3 for BRM rewriting)
Current state: S_1
Next state: S_0
Event: (N>0) & (BN_{BRM}=0)
Actions: rewriting all queue turned around BRM cells in queue from destination to source with the contents of this BRM cell;
Queue this BRM cell to source;
N:=N+1; (lock N)

T_{12} (related to destination 3 for BRM dropping)

Current state: S_1
Next state: S_0
Event: (N>0) & & ($BN_{BRM} - 0$)
Actions: drop all queue turned around BRM cells in queue from destination to source;
Queue this BRM cell to source;
N:=1; (lock N)
T_{13}(related to destination 3 for BRM cell with $BN_{BRM=1}$)
Current state: S_1
Next state: S_0
Event : (N>0) & & (BN_{BRM}=1)
Actions: queue this BRM cell to source; (destination generated BRM cell with BN_{BRM}=1 does not alter the queue from destination to source)
N:=N+1;
T_{14}(destination implicit)
Current state: S_1
Next state:S_0
Event: N=0; (queue is empty)
Actions: N:=1; (lock N)
Queue this BRM cell to source;

7. Scheduler machine (Source or destination behavior)

This is the primary section of the PEFSM model of the entire source or destination behavior where it has had to use our understanding and interpretation of the english specification and create a new and novel machine to represent the correct operation of the protocol. It arrives at a minimum representation adequate to have reasonably correct behavior of the ABR service. The scheduler is a machine that is intentionally left to implementation by vendors according to their needs and expertise, to allow for vendor differentiation. However there is a minimal requirement of the scheduler which is that cells from the VC (RM cells as well as data cells) are not transmitted at a greater rate than ACR. BY this it mean that the time between successive transmission of the cells for VC is not less than 1/ACR. It has ensured that this is reflected in the formal specification, while not necessarily incorporating all of the implementation issues, including those relating to when there are multiple VCs awaiting transmission at the source on a cell time etc., which it feels is outside the essential scope of this formal specification. The source policy for ABR specifies the time at which different types of the cells are to be transmitted. FRM cells are transmitted upon a timer expiration so as to ensure that a oribe for the feedback information is transmitted with a minimum frequency. At the time an FRM cell is transmitted based on the timer expiration, actions specified by the source policy modify the allowed rate, as well as other variables. To avoid having a complex interface between the source protocol state machine and the scheduler machine, we have partitioned the state transitions machines relating to the source policy between the two machines. Thus the number of shared variable between the state machine is kept small only relating to the state of the each of the queue. In fact the queue of the data cells is entirely within the preview of the scheduler machine alone, since all of the actions that related to the data cell transmissions are represented in the transitions within the scheduler. The scheduler machine keeps track of time , which is the time since the transmission of the FRM cell, as also the time since the transmission of the last FRM cell, as also the time t between successive transmission of the cells of the VC, which are successive separated by time 1/ACR. Note, however that these variables are local to the scheduler machine. The variable ACR between the scheduler and the source, the counter for the size of the queue of BRM cells and the number of FRM cells sent by the scheduler since the last BRM cell was received by the source feedback from the network are shared variables that have to be locked for updates the shared variable that is ACR in the negligible time, the impact of the locking by the source is minimal- the scheduler has essential no delays when it needs access to the variable ACR or the queue size of the BRM cell queue. This allows both machines to have current access to the variables with no performance impact. The timer T is used to track the amount of the time since the last FRM cell was transmitted by Th e scheduler. Upon expiration of the time T, T>=Trm, if at least Mrm in rate cells have been sent then the scheduler keeps tracks of the number of the turned around BRM cells in the queue to be transmitted in the shared variable X, and the number of FRM cells transmitted by the scheduler since the last FRM cell was received in the shared variable Y. To ensure that successive cells for VC are sent after at least a timer period of 1/ACR the scheduler uses a local timer t. when it is time to transmit an FRM cell, we model the source behavior as the scheduler creating an FRM cell with the appropriate fields and transmitting it. Thus, FRM cells are not actually queued between the source and the scheduler. The queue of the data cells is a virtual one , since we do not model the higher level layer function that hands down the data cells to the ARB protocol machines. We model it simply by the scheduler recognizing the event E when a data cell is available to be transmitted.

Because of the way we model the interfaces between the destination-source-scheduler machines, we ensure that the turned around BRM cell arrives via the source machine. It is believed that this is the correct way to describe what typically happens in an end system when there is received information that has to be turned around and transmitted.

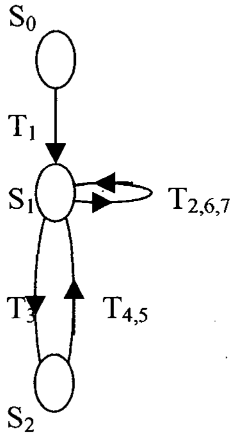


Figure: scheduler machine

It is the best to interface the receiving portion to the sending end of the link via the source protocol machine to keep the interfaces simple. Transitions T_2 and T_3 combine the conditions specified by the rules that related to the transmission of an FRM cell. The predicates for the transmission T_2 specify when an FRM cell may be transmitted. If the sources rules are enabled that is transmission T_3's predicate is satisfied, it moves to a state S_2 and the current rate ACR is set to the initial rate ICR.

In transition T_5, a rate change may occur at the time an FRM cell has to be transmitted, reflecting the actions specified. Specially if the counter Y exceeds a threshold value Crm then a rate reduction occurs, where Y is number of FRM cells queued to the scheduler since the last BRM cell with $BN_{BRM}=0$ is received. If we think of an 'imaginary cycle' of transmission as being an FRM cell and (Nrm-1) cells comprising both turned-around BRM cells and data cells, the source policy provides a dynamic priority for the transmission of RM cells and data cells within the cycle. Subsequent to transmission of the FRM cell at the beginning of this imaginary cycle, the source policy provides priority to transmit a queued turned-around BRM cell, prior to transmission of data cells. Subsequent to the transmission of the turned-around BRM cell, data cells have priority, until Nrm – 2 data cells have been transmitted. This broad principle, as specified in rules 3(b) and 3(c) in the source policy is reflected in transitions T_6 and T_7 in the formal specification.

Transitions T_2, T_6, and T_7 transmit the appropriate cell when the local timer t has exceeded the minimum time 1/ACR, and there are cells to be sent. The corresponding counters are modified and t reset to zero.

The transition diagram of the scheduler machine is in Fig. 4.

8. Scheduler machine

8.1 States

S_0 : initial state

S_1 : active state

S_2 : state in which rate may be changed

8.2 Variables

X_1 : number of BRM cells transmitted by Scheduler since last FRM cell transmitted by Scheduler

X_2 : number of data cells transmitted by Scheduler since last FRM cell transmitted by Scheduler

X : number of BRM cells in queue from Source to Scheduler / * shared variable with Source * /

Y : number of FRM cells sent by Scheduler after initialization and before receiving the first BRM cell with BN = 0 or since last BRM cell with $BN_{BRM} = 0$ received by Source /* shared variable with Source * /

8.3 Timer

t: time elapsed since last cell sent by Scheduler

T: time elapsed since last FRM cells sent by Scheduler

8.4 Events

E: data cells waiting to be sent

8.5 Transition

T_1 (Source 1)

current state: S_0

next state: S_1

event:

actions:

$X_1 := 0;$

$X_2 := 0;$

$T : = 0;$

$t := 0;$

send an FRM cell;

$Y := 0;$

T_2 (Source 3a and 5)

current state: S_1

next state: S_2

event:

$(((X_1 + X_2 \geq \mathrm{Mrm})\ \&\ \&\ (T \geq \mathrm{Trm}))\ \mathrm{ll}$

$(X_1 + X_2 = \mathrm{Nrm} - 1))\ \&\ \&\ ((\mathrm{ACR} \leq \mathrm{ICR})\ \mathrm{ll}$

$(T \leq \mathrm{ADTF}))\ \&\ \&\ (t \geq 1/\mathrm{ACR})$

actions:

T_3 (Source 3a and 5)

current state: S_1

next state: S_2

event:

$(((X_1 + X_2 \geq \mathrm{Mrm})\ \&\ \&\ (T \geq \mathrm{Trm}))\ \mathrm{ll}$

$(X_1 + X_2 = \mathrm{Nrm} - 1))\ \&\ \&\ (\mathrm{ACR} > \mathrm{ICR})\ \&$

$\&\ (T > \mathrm{aDTF})\ \&\ \&\ (t \geq 1/\mathrm{ACR})$

actions:

ACR := ICR; / * lock ACR */

T_4 (Source 3a 5, 6 and 7)

current state: S_2

```
    next state: $S_1$
    event:
        Y < Crm
    actions:
        $CCR_{FRM}$ := ACR;
    send an FRM cell;
    Y := Y+1; /* lock Y */
    $X_1$ := 0;
    $X_2$ := 0;
    t := 0;
    T := 0;
$T_5$ (Source 3a 5, 6 and 7)
    current state: $S_2$
    next state: $S_1$
    event:
        Y < Crm
    actions:
        ACR: ≤ ACR * (1-CDF);
        ACR := max {MCR, ACR}; /* lock ACR */
        $CCR_{FRM}$ := ACR;
        Send an FRM cell;
        Y := Y+1; /* lock Y */
        $X_1$ := 0;
        $X_2$ := 0;
        t := 0;
        T := 0;
$T_6$ (3(b) send a BRM cell to network)
    current state: $S_1$
    next state: $S_1$
    event:
        (X>0) & & (($X_1$ + $X_2$ < Mrm) ll (T < Trm))
        & & ($X_1$ + $X_2$ < Nrm – 1) & & (($X_1$ = 0) ll
```

!E) & & $(t \geq 1/ACR)$

actions:

$X := X-1$;/*lock X */

$X_1 := X_1 + 1$;

Send a BRM cell;

$t := 0$;

T_7 (Source 3(c) send a data cell to network)

current state: S_1

next state: S_1

event:

E & E $((X=0) \| (X_1 > 0))$ & & $((X_1 + X_2$

$< Mrm) \| (T<Trm))$ & & $(X_1 + X_2 < Nrm$

$- \quad 1)$ & & $(t \geq 1/ACR)$

actions:

$X_2 := X_2 + 1$;

send a data cell;

$t := 0$;

9. Conclusion

In this paper, we have described a parameterized communicating extended finite state machine model for the ATM Forum's ABR service specification. While it is often appropriate to develop a congestion control mechanism and the protocol through English descriptions and simulations etc., it is valuable to then derive a formal model based on that description. This allows for an unambigous interpretation of the specification by implementers to allow for an efficient implementation and for interoperability between implementations. The formal specification would also facilitate formal analysis, such as validation, verification, and testing, of the ABR protocol. Based on the formal specification, preliminary results have been obtained on the correctness and performance of the ABR protocol [7].

We believe that the formal model for the source, destination, and the scheduler developed here clarifies what appears on the surface to be a complex English specification. We have introduced the concept of a scheduler (that was only implicit in the English specification), and two simple interfaces among the source, destination, and the scheduler machines. Using these interfaces, and interpreting the English description of the source and destination rules, three corresponding PEFSM's have been developed. The novelty is in the interfaces we have introduced among the three machines, including the recognition among the three machines, including the recognition that that the destination protocol machines interfaces only to the source protocol machine. Finally, the scheduler interfaces only to the source protocol machine.

10. Acknowledgements

Valuable comments from John Kenney and Huayan Wang on the earlier drafts of this paper are deeply appreciated. We also thank A. Udaya Shankar who was involved in the initial stages of this work. Melody Moh is supported in part by the NSF grant NCR-9706461, the SJSU Faculty Development Award, and collaboration grant funded by Silicon Valley Research.

11. Appendix : ATM forum traffic management specification V4.0

For reference, we include the English specification from [1] for the source, destination, and switch behavior.

Source behavior

The following items define the source behavior for CLP = 0 and CLP = 1 cell streams of a connection. By convention, the CLP = 0 stream is referred to as in-rate, and the CLP = 1 stream is referred to as out-of-rate, and the CLP = 1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

(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. The source may always send in-rate cells at a rate less than or equal to ACR.

(2) Before a source sends the first cell after connection setup, it shall set ACR to at most ICR. The first in-rate cell sent shall be a forward RM-cell.

(3) After the first in-rate forward RM-cell, in-rate cells shall be sent in the following order:

(a) The next in-rate cell shall be a forward RM-cell if and only if, since the last in-rate forward RM-cell was sent, either:

(i) at least Mrm in-rate cells have been sent and at least Trm time has elapsed, or

(ii) Nrm – 1 in-rate cells have been sent.

(b) The next in-rate cell shall be a backward RM-cell if condition (a) above is not met, if a backward RM-cell is waiting for transmission, and if either:

(i) no in-rate backward RM-cell has been sent since the last in-rate forward RM-cell, or

(ii) no data cell is waiting for transmission.

(c) The next in-rate cell sent shall be a data cell if neither condition (a) nor condition (b) above is met, and if data cell is waiting for transmission.

(4) Cells sent in accordance with source behaviours #1, #2, and #3 shall have CLP = 0.

(5) Before sending a forward in-rate RM-cell, if ACR>ICR and the time T that has elapsed since the last in-rate forward RM-cell was sent is greater than ADTF, then ACR shall be reduced to ICR.

(6) Before sending an in-rate forward RM-cell, and after following behaviour #5 above, if at least CRM in rate forward RM-cells have been sent since the last

backward RM-cell with BN=0 was received, then ACR shall be reduced by at least ACR*CDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR.

(7) After following behaviours #5 and #6 above, the ACR value shall be placed in the CCR field of the outgoing forward RM-cell, but only in-rate cells sent after the outgoing forward RM-cell need to follow the new rate.

(8) When a backward RM-cell (in-rate or out-of-rate) is received with CI=1, then ACR shall be reduced by at least ACR*RDF, unless that reduction would result in a rate below MCR, in which case ACR shall be set to MCR. If the backward RM-cell has both CI=0 and NI=0, then the ACR may be increased by no more than RIF*PCR, to a rate not greater than PCR. If the backward RM-cell has NI=1, the ACR shall not be increased.

(9) When a backward RM-cell (in-rate or out-of-rate) is received, and after ACR is adjusted according to source behaviour #8, ACR is set to at most the minimum of ACR as computed in Source Behaviour #8, and the ER field, but no lower than MCR.

(10) When generating a forward RM-cell, the source shall assign values to the various RM-cell fields as specified for source-generated cells in Table 5-4.

(11) Forward RM-cells may be sent out-of-rate (i.e. not conforming to the current ACR). Out-of-rate forward RM-cells shall not be sent at a rate greater than TCR.

(12) A source shall reset EFCI on every data cell it sends.

(13) The source may implement a use-it-or-lose it policy to reduce its ACR to a value which approximates the actual cell transmission rate. Use-it-or-lose-it policies are discussed in Appendix I.8.

Notes:

(1) In-rate forward and backward RM-cells are included in the source rate allocated to a connection.

(2) The source is responsible for handling local congestion within its scheduler in a fair manner. This congestion occurs when the sum of the rates of the scheduler. The method for handling local congestion is implementation specific.

Destination behavior

The following items define the destination behaviour for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP = 0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate.

(1) When a data cell is received, its EFCI indicator is saved as the EFCI State of the connection.

(2) On receiving a forward RM-cell, the destination shall turn around the cell to return to the source. The DIR bit in the RM-cell shall be changed from

'forward' to 'backward', BN shall be set to zero, and CCR, MCR, ER, CI, and NI fields in the RM-cell shall be unchanged except:

(a) If the saved EFCI state is set, then the destination shall set CI=1 in the RM-cell, and the saved EFCI state shall be reset. It is preferred that this step is performed as close to the transmission time as possible:

(b) The destination having internal congestion may reduce ER to whatever rate it can support and/or set CI=1 or NI=1. A destination shall either set the QL and SN fields to zero, preserve these fields, or set them in accordance with I.371-draft. The octets defined in Tables 5-4 as reserved may be set to 6A (hexadecimal) or left unchanged. The bits defined as reserved in Tables 5-4 for octet 7 may be set to zero or left unchanged. The remaining fields shall be set in accordance with Section 5.10.3.1 (note that this does not preclude looping fields back from the received RM-cell).

(3) If a forward RM-cell is received by the destination while another turned-around RM-cell (on the same connection) is scheduled for in-rate transmission:

(a) It is recommended that the contents of the old cell are overwritten by the contents of the new cell;

(b) It is recommended that the old cell (after possibly having been over-written) shall be sent out-of-rate; alternatively the old cell may be discarded or remain scheduled for in-rate transmission;

(c) It is required that the new cell be scheduled for in-rate transmission.

(4) Regardless of the alternatives chosen in destination behaviour #3 above, the contents of an older cell shall not be transmitted after the contents of a newer cell have been transmitted.

(5) A destination may generate a backward RM-cell without having received a forward RM-cell. The rate of these backward RM-cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connection. When a destination generates an RM-cell it shall set either CI=1 or NI=1, shall set BN=1, and shall set the direction to backward. The destination shall assign values to the various RM-cell fields as specified for destination generated cells in Tables 5-4.

(6) When a forward RM-cell with CLP=1 is turned around it may be sent in-rate (with CLP=0) or out-of-rate (with CLP=1).

Notes:

(1) 'Turn around' designates a destination process of transmitting a backward RM-cell in response to having received a forward RM-cell.

(2) It is recommended to turn around as many RM-cells as possible to minimize turn-around delay, first by using in-rate opportunities and then by using out-or-rate opportunities as available. Issues regarding turning RM-cells around are discussed in Appendix I.

Switch behaviour

The following items define the switch behaviour for CLP=0 and CLP=1 cell streams of a connection. By convention, the CLP=0 stream is referred to as in-rate, and the CLP=1 stream is referred to as out-of-rate. Data cells shall not be sent with CLP=1.

(1) A switch shall implement at least one of the following methods to control congestion at queuing points:

(a) EFCI marking : The switch may set the EFCI state in the data cell headers;

(b) Relative Rate Marking: The switch may set CI=1 or NI-1 in forward and /or backward RM-cells;

(c) Explicit Rate Marking: The switch may reduce the ER field of forward and/or backward RM-cells (Explicit Rate Marking);

(d) (VS/VD Control): The switch may segment the ABR control loop using a virtual source and destination.

(2) A switch may generate a backward RM-cell. The rate of these backward RM-cells (including both in-rate and out-of-rate) shall be limited to 10 cells/second, per connection. When a switch generates an RM-cell it shall set either CI=1 or NI=1, shall set BN=1, and shall set the direction to backward. The switch shall assign values to the various RM-cell fields as specified for switch-generated cells in Tables 5-4.

(3) RM-cells may be transmitted out of sequence with respect to data cells. Sequence integrity within the RM-cell stream must be maintained.

(4) For RM-cells that transit a switch (i.e. are received and then forwarded), the values of the various fields before the CRC-10 shall be unchanged except;

(a) CI, NI, and ER may be modified as noted in #1 above,

(b) RA, QL, and SN may be set in accordance with I.371-draft.

(c) MCR may be corrected to the connection's MCR if the incoming MCR value is incorrect.

(5) The switch may implement a use-it-or-lose-it policy to reduce an ACR to a value which approximates the actual cell transmission rate from the source. Use-it-or-lose-it policies are discussed in Annex F.

Notes:

1. A switch queuing point is a point of resource contention where cells may be potentially delayed or lost. A switch may contain multiple queuing points.

2. Some example switch mechanisms are presented in Apendix I.

3. The implications of combinations of the above methods is beyond the scope of this specification.

References

1. J.A.S.Monteiro, M.Gerla, Leaky Bucket input rate control in ATM networks, ICCC 1990, New Delhi, India, 1990, pp,. 370-376

2. K.Narendra, M.Thathachar, Learning automata : a survey, IEEE Trans. SMC 4(4) (1974) 323-334.

3. E.P. RATHGEB, Modeling and performance comparison of policing mechanisms for ATM networks, IEEE J. SAC (April 1991) 325-334.

4. A.V.Vasilakos, G.Papadimitrios, A new approach to the design of reinforcement scheme for learning automata: Stochastic Estimator Learning Algorithms, Neuro-computing 7(3)(1995) 275-297.

5. A.V.Vasilakos, C.T.Paximadis, Fault-tolerant routing algorithms using SEDEL automata for high-speed packet switched networks, IEEE Trans. Reliability (December 1994) 582-593.

6. A.V.Vasilakos, N.H.Loukas, A.F.Atlasis, The use of learning algorithms in the ATM networks call admission contril problem: a methodology, ICCC'95, Seoul, Korea, August 1995.

7. M.A.L.Thatcachar, P.S.Satry, A class of rapidly converging algorithms for learning automata, IEEE Trans. SMC 15(1) (1985).

8. A.V.Vasilakos, G.I.Papadimitriou, Ergodic discretized estimator learning automata with high accuracy and high adaptation rate for non-stationary environments. Neuro-computing 4 (1992) 181-196.

9. F.Guillemin, P.Boyer, A basic requirement for the policing function in ATM networks : asymptotic analysis and simulations, submitted for publication.

10. ATM Forum Traffic Management Specification Version 4.0 af-tm-0056, April 1996, available as : ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.pdf.

11. L.Roberts, Enhanced PRCA (Proportional Rate-Control Algorithm), ATM Forum Contribution, AF-TM 94-0735RI, August 1994.

12. D.Lee.K.K.Ramakrishshna, W.M.Moh, A.U.Shankar, Performance and correctness of the ATM ABR rate control scheme, Proc. IEEE INFOCOM'97,April 1997.

13. P.Mars, W.J.Poppelbaum, Stochastic Determination Average Processors, Peter Peregrinus, Stevenage, UK, 1981.

14. D.Anick, D.Mitra, M.M.Sondhi, Stochastic theory of a data-handing system with multiple sources, Bell System Tech.J.61(8)(1982) 1871-1894.

15. A.V.Vasilakos, G.I.Papadimitriou, Ergodic discretized estimator learning automata with high accuracy and high adaptation rate for non-stationary environments, neuro-computing 4(1992) 181-196.