Traffic Policing in ATM Networks using Learning Automata

Y.Seifi[1], M. R. Meybodi[2]
[1]*University of Bualisina, Faculty of Engineering, Hamadan, Iran*
[2]*Amirkabir University of Technology, Faculty of Computer Engineering, Tehran, Iran*

Abstract: Different approaches have been suggested for Traffic policing in an ATM network which majors are windowing algorithm and Leaky Bucket(LB). Windowing algorithms do well in packet networks but are unsuitable for high speed networks such as ATM.
In this article a new group of traffic policing method is suggested which at each moment improves the performance of policy by deliberating source attitudes or changing different parameters of leaky bucket. This group of policies deliberates sources attitude or efficiency of using sources by learning automata, and then improves the efficiency.
Results of simulation shows that new approaches are able to reduce the percentage of cell loss and finally improve the performance of using sources , this group of methods are suitable for explosive traffic sources which there is no precise representation of their traffic properties or some kinds of traffic sources which estimating their properties is complex.
Keywords: Asynchronous Transfer Mode, Learning Automata, Leaky Bucket, Usage control parameter, traffic policing

1. Introduction

ATM networks have their own weaknesses and features. One of their weaknesses is handling of congestion and buffer overflow especially when traffic sources are different with each other.

One of their major capabilities is supporting kinds of services with variant rate and various quality of service (QOS). For avoiding congestion, call admission control (CAC) methods are used at the time of establishing connection. These methods determine whether a connection can be accepted or not. CAC methods by monitoring current network load, new connection specification and allowed quality of service, try to use network resources as well as possible and prevent from congestion. After accepting a connection it is the duty of traffic policing mechanism to control applicability of source function with primary negotiation parameters [1-3].

Traffic policing would be more efficient if negotiation has been done with more detailed parameters. Meanwhile if attributes of negotiation become very complex, usage parameter control would be more difficult and it is clear that in many cases such as video services finding correct and precise attributes of service is impossible.

Various methods for traffic policing in ATM networks are suggested which majors of them are windowing algorithm and leacky bucket. Windowing algorithms operate in packet networks very good but are not suitable for high speed networks such as ATM.

In [4] leacky bucket has been introduced as the most efficient policing method. In This mechanism by policing and shaping inbound network traffic rate current quality of service is protected. This method is very simple and can be implemented in hardware. Meanwhile because of statistical nature of the burst, when leacky bucket mechanism is used for policing, bursty sources have some errors in estimating attributes of the source, such as average rate. These errors increase when periods of estimation reduce. On the other hand increasing periods of estimation can decline dynamic nature of an ATM network [5]. In an ATM network, parameter control always has a major difficulty, because on one side, finding suitable estimation requires a long time, and on the other side, minimizing changes and errors requires a short time [6]. So policing tools which are suggested until now can not police bursty sources efficiently. For avoiding these difficulties two groups of policing mechanisms have been suggested, the former can recognize source behavior and the latter can adjust various policing parameters.

Suggested algorithms in [7-8] are some policers in the first group which determine whether behavior of sources is according to the negotiated parameters or not, by monitoring instant behavior of the source. When source behavior is normal its cells pass from policer without any changes but if source's behavior isn't normal and doesn't consider negotiated specifications it's cells must be lost.

In the second group of policers, various parameters of leacky bucket are adjusted so that functionality of the policer becomes better and decreases number of lost cell. Our policers in this article are like this kind, which controls parameters using learning automata.

In the second group, parameters are adjusted by learning automata. Vasilakos and Atlasis have proposed some traffic policers in ATM networks using learning automata in 1994 and 1997. The former which is done in 1994 uses a learning automata called SELA, for traffic policing of bursty sources. Simulations of this algorithm in [9] shows that new method causes a considerable optimization in performance of leacky bucket.

In another policer which is proposed by Vasilakos, equipped policer with learning automata is placed in internal nodes of the network. Operation of this method is such as previous policer designed by this researcher[9].

In this article three algorithms based on learning automata are proposed for traffic policing in ATM networks.

First algorithm in comparison with proposed algorithm by Vasilakos and Atlasis is different from two aspects. First, the used learning automata is different. Second, environment response of our learning automata is number of cell losses in the last time interval which is different from the considered response by Vasilakos and Atlasis.

Second algorithm is such as first one, but in any run of learning automata bucket size changes in spite of leack rate parameter.

Third algorithm in spite of SELA uses L_{rep} automata which is more simple, faster and its performance is comparable with SELA.

2. Simulations

Simulations are done by ATM/HFC(Hybrid Fiber Coax) simulator which is produced in NIST [10]. This simulator can evaluate ATM and HFC networks. Meanwhile provides an interactive modeling environment with a graphical user interface for users. NIST has produced this tool using C programming language in X windows environment which runs in Unix platform. We have used this simulator to examine our algorithms on a specific traffic source and network topology.

3. Results

Results of simulations shows that our proposed algorithms can reduce number of cell losses in network and revise performance of network resources. In some cases cell loss reduces twenty times. This group of policers are suitable for bursty traffic sources which finding specification of them is not impossible.

4. References

1) CCITT, Recommendation I.121 : “Broadband aspects of ISDN”, Blue Book,, Vol. III. 7, Geneva, Switzerland, 1989.

2) E. P. Rathgeb, “Modeling and performance comparison of policing mechanisms for ATM networks”, IEEE J-SAC April 1991 pp 325-334.

3) A.I Elwalid, D. Mitra, “Analysis and design of rate-base congestion control of high speed networks, I:stochastic fluid models, access regulation”, Queueing Systems 9, 1991, pp. 29-63.

4) Milena Butto, Elisa Cavallero and Alberto Tonietti, “Effectiveness of the leacky bucket policing mechanism in ATM networks”, IEEE. J. Select. Areas. Commun., vol. 9,no. 3, pp. 335-342 April 1991.

5) Irfan Khan, et al, “Traffic control in ATM networks”, Computer Networks and ISDN System 27(1994)85-100.

6) Z. Jiang, Z.liu, “An improved algorithm of usage parameter control in ATM networks”, ICCT’96, vol.1 pp. 24-27.

7) G. Gallassi, G. Rigolio,L. Fratta, “ATM: Bandwidth assignment and bandwidth enforcement policies”, IEEE GLOBECOM, pp.1788-1793, 1989.

8) J. A. Monteiro, M Gerla,L. Fratta, “Leacky bucket input rate control in ATM networks”,ICCC 1990, New Delhi, India, pp. 370-376, 1990.

9) A. V. Vasilakos, A. F. Atlasis, “Effectiveness of the LB-SELA policing mechanism in an ATM network node”, IEEE GLOBECOM’95, pp. 627-631, 1995.

10) N. Golmie, F. Mouveaux, L. Hester, Y. Saintillan, A. Koenig, D. Su, “The NIST ATM/HFC network simulator version 4.0”, high –speed networks technology group, advanced networks technology division, information technology laboratory. NIST, December 1998.