TCP TRAFFIC SHAPING in ATM NETWORKS

İbrahim Koçyiğit
Faculty of Engineering and Architecture, Department of Electronics
Uludağ University, Görükle, Bursa-TURKEY
E-mail : kocyigit@uludag.edu.tr

Keywords; *ATM, Traffic Shaping, VSA Algorithm, GCRA*

ABSTRACT

In this paper, some simulation results concerning the performance of the TCP protocol implemented in high-speed ATM networks are presented. A simple ATM network is concerned, comprising four TCP connections share the bandwidth of a bottleneck link with some background traffic. In our simulation scenario, the background traffic level and node buffer size are taken as a variable parameter. Both the background traffic and the TCP traffic are shaped according to the GCRA algorithm. The effect of the background traffic on the TCP protocol performance is studied, varying for node buffer size and background load.

I. INTRODUCTION

In the future, the data traffic is expected to be a relevant part of the load in ATM networks. That is, the TCP will be the most frequently used transport protocol in the ATM environment . Our work concentrates on the effect that the heterogeneous traffic present in the network, that we call background traffic, may have on the TCP performance. The importance of the presence of background traffic goes beyond the reduction of the bandwidth available to TCP, since background traffic interferes with the TCP behavior by altering the probability of cell losses within node buffers. The results presented in this paper are obtained via simulation, implementing a TCP layer within an ATM network simulator named CLASS (ConnectionLess ATM Services Simulator). CLASS[3] was recently developed at Politecnico di Torino-Italy for the performance study of data services in ATM networks.

II. TRAFFIC SHAPING

The user of the services offered by the ATM network may perform some kind of control on the traffic it generates in order to verify that it sticks to the negotiated parameters and will thus not be rejected by the network. This operation is clearly very similar to the traffic policing, but it is preventive and not repressive and is generally called traffic shaping, because the operation performed is aimed at the modification of the traffic characteristics, generally by smoothing its burstiness, i.e. to give a pre-defined shape to its profile. Traffic shaping has three main purposes. These are:

1. The characteristics of a flow of cells that has endured a shaping process are much easier to be described, thus the shaping of the traffic allow the user to negotiate the transmission parameters more easily.
2. If the shaping algorithm is known to the network too, it helps also the network in the management of the call acceptance because it can more easily predict the behavior of the generator.
3. Monitoring of the traffic on the network side, i.e. traffic policing, is much easier and reliable if the input flow has known characteristics.

A number of shaping policies and algorithms have been proposed in literature. But, each one have some advantages and drawbacks. The traffic control algorithm that has been chosen for the inclusion in this study is the Virtual Scheduling Algorithm, described in the recommendation I.371 of ITU[2]. This algorithm is basically a token bucket . In any case, although the basic algorithm is the same the actions that are taken by a shaper are radically different from those taken from a policer (``bad" cells are delayed and not discarded). Each time a cell becomes available for transmission, the Virtual Scheduling Algorithm determines whether the cell is conforming with the Traffic Contract of the connection. The VSA not only provide to control the traffic characteristics, but also provide a means for the formal definition of traffic conformance to the Traffic Contract. The VSA requires only the definition of two parameters: the time increment between cells T and the time limit τ . The time increment between cells is clearly the time that should pass between two consecutive cells if the traffic was generated at constant bit rate. τ is the time jitter that is allowed and is clearly related to the parameter generally called ``cell delay variation". The VSA keeps track of a Theoretical Arrival Time (TAT ), which is the ``nominal" arrival time of the next cell assuming equally spaced cells when the source is active. If the actual arrival time of a cell is not ``too" early relative to the TAT , in particular if the actual arrival time is after TAT time of the first cell be t_a (1). In this case the cell is transmitted immediately and the TAT is initialized to the t_a (1) + T . For all subsequent cells one of the following three alternatives is given:

1. The arrival time t_a of the cell is greater than or equal to TAT ; then the cell is compliant, it is transmitted immediately and TAT is updated to the value t_a + T .
2. The cell arrives at time t_a with

$$TAT-\tau \leq t_a \leq TAT,$$

The cell is again conforming, and the TAT is increased again by the increment T:

$$TAT=TAT+T$$

3. The arrival time of the cell is smaller than TAT-τ, then the cell is non_conforming; the cell transmission time t_t is set to actual value of TAT and TAT is updated to the value TAT+T.

In practice the TAT of the next cell is always set to the present cell transmission time, plus the time increment, i.e.,

$$TAT\ (k + 1) = t_t\ (k) + T:$$

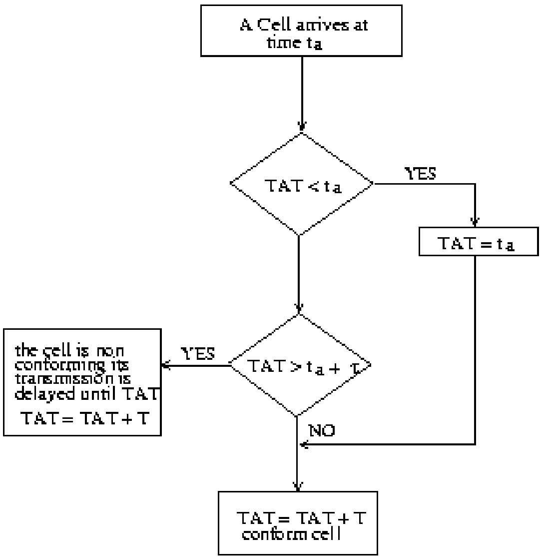

Fig 1. Flow chart of the VSA

Given a shaping algorithm there are several possibilities for its implementation. One of the choices that must be made concerns the level of multiplexing before the shaping is performed. The easiest solution is to perform the shaping at the front end on the transmitter, negotiating the traffic parameters for all the traffic generated by the user .The shaping policies applied to the users can be classified into two categories.

VP-based shaping policy ; The users grouped under this category perform the shaping of the traffic at the interface between the user buffer and the transmission link. This means that these users have only a single ``traffic shaper", that operates upon the whole traffic generated by the user, without considering the destination, VCI or traffic relation.according to this, there is only one group and the output multiplexer is not present, all the VC are grouped together and the multiplexing among the VCs is done at the message level, i.e. all the cells of the same message are in any case transmitted sequentially one after the other without interleaving with cells of other messages. The user negotiates the bandwidth needed for the transmission of the whole traffic with the node, and sets accordingly the traffic parameters. The other parameter needed by the VSA, namely T, is computed as follows:

$$T = C / B_w .\beta$$

C = Link capacity
B_w = Mean bandwith required by the user
β = Bandwidth allocation factor

VC-based shaping policy; The users grouped under this category perform the shaping of the traffic on a VC per VC basis. This means that there are as many ``traffic shapers" as the number of VCs. According to this, there are N groups, where N is the number of VCs (traffic relations) generated by the user, the output multiplexer is present, while the group multiplexers are not present since each group contains only one VC. Within this VC per VC shaping policy the multiplexing stage may have a considerable impact on the cell delay jitter. In this category, T can be calculated as above.

III. SIMULATION SCENARIO and NUMERICAL RESULTS

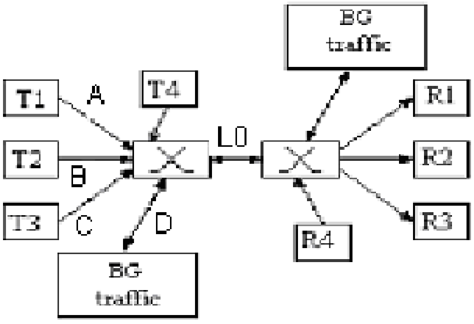

Fig 2. The simulated ATM network topology

We consider a very simple network, sketched in Fig. 2, comprising only two ATM switches. The data rate on each channel is 150 Mbit/s, and channel L0, linking the two ATM switches, is the system bottleneck. Four unidirectional TCP connections share the network resources with a variable amount of background traffic. The background traffic messages are generated according to a Poisson process, with a truncated geometric message length distribution with mean equal to 25 cells and maximum length 200 cells. The burstiness of both the TCP connections and the background traffic can be controlled with a shaping device that operates according to VSA(an adaptation of the GCRA) recommended by ITU-T . When the background traffic is shaped, we assign to each connection $\beta = 1.2$ and $\tau = 0$. All of TCP transmitters have a maximum window size equal to 60 segments. TCP protocol always transmits segments of the the maximum possible size (9180 bytes). Numerical results are presented as curves referring to two performance indices:

1. The useful throughput which is called goodput, at the TCP receivers, obtained considering the received data, but discarding all the faulty and the retransmitted segments;
2. The efficiency of the TCP connections, i.e., the ratio between the goodput and the total offered load of TCP connections.

The performance of this very simple ATM network was studied as a function of three variables: the background traffic load and traffic characteristics the TCP traffic shaping parameters and node buffer size. The results presented in Fig.3, 4, 5 and 6 are obtained when all the links A, B, C, D in Fig. 2 have length 500 km, while the links from the second ATM switch to the TCP receivers are assumed to have negligible length. The TCP connection length is thus 1000 km. The results are presented as a function of the background traffic load, for two values of the node buffer size in front of the congested link, L0, 1000 and 5000 cells, with the background traffic shaped. As expected, when no shaping is performed on the TCP traffic the TCP goodput steadily decreases with increasing background traffic. An increase in the node buffer size results in an increase of the TCP goodput, even if this increase is not very significant(see and compare fig 3. and fig 5.)

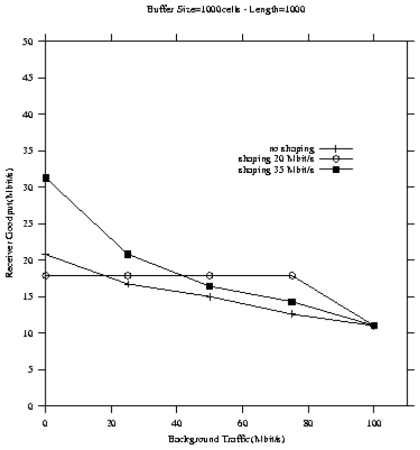

Fig 3. Average receiver goodput of TCP connections for the simulated scenario (Buffer size =1000 cells and BG is shaped)

The results when the traffic on TCP connections is shaped are presented on the same charts with the circle and square markers for the cases of 20 and 35 Mbit/s shaping, respectively, assuming $\tau = 0$. The results show that smoothing the burstiness of the traffic offered to the network allows TCP connections to better exploit the available resources.

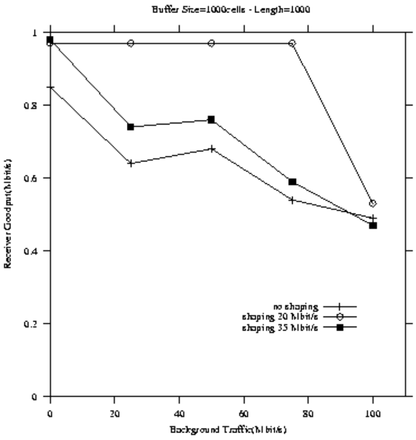

Fig 4. Average receiver efficiency of TCP connections for the simulated scenario (Buffer size = 1000 and BG is shaped)

The shaping values correspond to 7/30 and 2/15 of bottleneck link capacity. In particular, when a 35 Mbit/s shaping is enforced on TCP connections and no background traffic is present, the TCP connections completely saturate the link capacity. In any case, the goodput achieved with a 35 Mbit/s shaping is always greater than the unshaped goodput, regardless of the node buffer size and the background load.

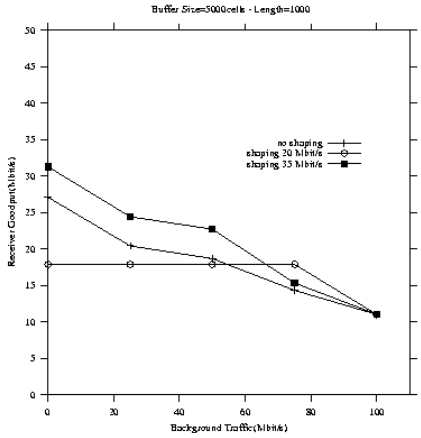

Fig 5. Average receiver goodput of TCP connections for the simulated scenario (Buffer size = 5000 cells and BG traffic is shaped)

The situation is slightly different when we analyze the curves with 20 Mbit/s shaping. In this case the TCP goodput is limited by the shaping function, not by the window mechanism, and it remains constant until the background load is increased to 75 Mbit/s. In this case, for high background traffic load, the goodput is greater than the one obtained in the case without shaping and with 35 Mbit/s shaping. It is interesting to notice that in the case of 100 Mbit/s background traffic load, when the network is clearly overloaded, the performance of TCP connections is basically independent from the shaping.

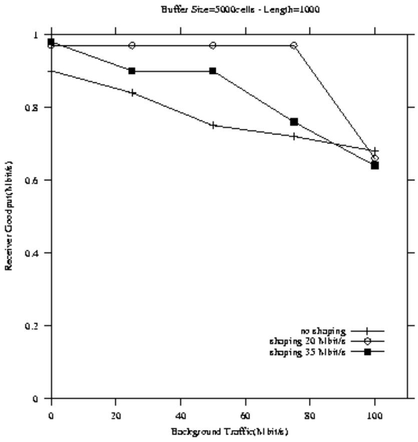

Fig 6. Average receiver efficiency of TCP connections for simulated scenario (Buffer size = 5000 and BG traffic is shaped)

The curves in figure 4 and 6 show that as soon as the total traffic offered to the network exceeds the bottleneck capacity, the efficiency of TCP protocol becomes very poor. Moreover, the more bursty is the traffic, the poorer is efficiency.

CONCLUSIONS

Numerical results clearly showed that shaping the traffic on the TCP connections greatly improves the TCP performance. Shaping can be applied with no modification of the TCP protocol itself, but it implies the negotiation of a traffic contract between the network and the TCP protocol entities at connection setup.

REFERENCES

1.“ATM Forum Traffic Management Specification 4.0”, ATM Forum Technical Committee Traffic Management Working Group, April 1996.
2.CCITT Recommendation I.371, “Traffic Control and Congestion Control in B-ISDN ”, 1993.
3.“CLASS – The User Manual”, Politecnico di Torino, Dipartimento di Elettronica, Italy, May 1998.
4.KOÇYİĞİT, İ., “Farklı QoS Sınıflarını Destekleyen ATM Şebekeler İçin Bir CAC Algoritması ”, ELECO 2000, 8-12 Kasım 2000, Sagem – Bursa.
5.MARSAN, M., BIANCO, A.,CIGNO, R., MUNAFO, M., “Some Simulation Results About Shaped TCP Connections in ATM Networks”, Italy, December 1995.
6.GORALSKI, Walter J., “Introduction to ATM Networking”, McGraw-Hill,1995.
7. Akdeniz ,F. , “ Olasılık ve İstatistik ”, Ankara Ü. Fen . Fak. Yayınları, Ankara 1994.