A Method for Traffic Scheduling Based on Token Bucket QoS Parameters

Conference Paper · April 2001

Source: CiteSeer

CITATIONS
0

READS
42

2 authors:

Fernando Moreira
Portucalense University
110 PUBLICATIONS **143** CITATIONS
SEE PROFILE

José Ruela
Institute for Systems and Computer Engineering of Porto (INESC Porto)
53 PUBLICATIONS **441** CITATIONS
SEE PROFILE

A Method for Traffic Scheduling Based on Token Bucket QoS Parameters

Fernando Moreira[1] José Ruela[2,3]

Departamento de Informática, Universidade Portucalense, Porto, Portugal (fmoreira@upt.pt)

DEEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal (jruela@inescporto.pt)

Abstract

This paper proposes and describes a new method that can be used to schedule individual traffic flows, which share a common channel and have been shaped/policed by means of a Token Bucket (TB) mechanism. The method can also be extended to dynamically allocate bandwidth to this channel, based on information derived from TB states. Simulation results are presented and discussed.

I. INTRODUCTION

The advances in fast packet switching technologies, fostered by the deployment of ATM networks and the advent of IP networks based on gigabit technology, has considerably changed the service paradigm of packet networks.

The traditional best effort service model is being extended by models that support the negotiation and provision of predictable, measurable and differentiated Quality of Service (QoS), both for individual and aggregate traffic flows.

The ATM Forum defined a number of service categories based on a small set of traffic and QoS parameters: Constant Bit Rate (CBR), real time and non real time Variable Bit Rate (rt-VBR, nrt-VBR), Available Bit Rate (ABR), Unspecified Bit Rate (UBR) and Guaranteed Frame Rate (GFR) [1]. Relevant traffic parameters are the peak cell rate (PCR), the sustainable cell rate (SCR) and the maximum burst size (MBS), while the main QoS parameters are the cell delay, the cell delay variation (jitter) and the cell loss ratio.

Shaping and policing of VBR traffic characterized by the triplet (PCR, SCR, MBS) may be performed by a combination of a Leaky Bucket (for PCR enforcement) and a Token Bucket (TB) [2].

A TB is a non-negative counter which accumulates tokens at a constant rate r until the counter reaches its capacity b. A TB may be used to shape/police packet or cell based flows. A cell is said to be conforming (and therefore eligible for transmission) when it can claim a token from a non empty TB, in which case the counter is decremented. When the TB is empty, cells are either dropped or queued until the TB accumulates enough tokens. In this model a TB is parameterized by a token replenishment rate r and a bucket depth size b, such that:

$$r = SCR \quad (1)$$

$$b = MBS * (1 - SCR/PCR) \quad (2)$$

In ABR a source may negotiate a minimum cell rate (MCR) and request a maximum cell rate (PCR), while its actual rate is bounded by an allowed cell rate (ACR) explicitly indicated by the network. This rate is usually based on a max-min fair share of the available bandwidth on bottlenecked links. An interesting feature of ABR is that the sources receive feedback from the network to adapt their rates.

In IP networks, besides best effort, two models have been proposed so far: Integrated Services, with Guaranteed and Controlled Load classes [3] and Differentiated Services, with Assured and Expedited Forwarding classes [4]. The traffic specification of the Guaranteed class is based on TB parameters, as well.

In order to achieve end-to-end QoS in IP networks and to provide scalable solutions the current trend is to structure the transport network into an Edge and a Core network [5]. The Core network may be based on label switching techniques (such as Multiprotocol Label Switching) for streamlining the transport of flow aggregates according to QoS or traffic engineering policies. At the Edge of the network individual flows are subject to policing, classification and forwarding decisions based on service level agreements (SLA) and then mapped into appropriate core flows according to their QoS requirements.

This paper addresses some of the above problems, namely how to schedule individual flows, characterized by TB parameters, which will be transported as aggregate flows inside the network.

The paper is organized as follows. In Section II some well known scheduling algorithms are reviewed and a rationale for the method is presented in Section III. The new method is discussed in Section IV, and is evaluated by means of simulation in the fifth section. Finally, in Section VI some conclusions and directions for further research are derived.

II. RELATED WORK

Scheduling algorithms have been extensively described in the literature and a few examples are *First In First Out* (FIFO), *Round Robin* (RR), *Weighted Round Robin* (WRR), *Weighted Fair Queuing* (WFQ) [6], *Longest Queue First* (LQF), *Oldest Cell First* (OCF) [7], *Least Time to Reach Bound* (LTRB) [8]. Some of these methods use declared or measured rates in order to share the bandwidth among competing flows.

[1] Departamento de Informática, Universidade Portucalense, Rua Dr. António Bernardino de Almeida 541-619, 4200-072, Porto, Portugal.

[2] DEEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

[3] INESC Porto, Praça da República, 93 r/c, 4050-497, Porto, Portugal.

Packet fair queuing (PFQ) schemes have been proposed to approximate the idealized *Generalized Process Sharing* (GPS) [9] algorithm. A GPS server has N queues (each with a service share); during any time interval when there are M non empty queues, the M packets at the head of the queues are simultaneously served in proportion to their share. All PFQ schemes use the notion of a virtual time function - but differ on the choice of this function as well as on the packet selection policy.

WFQ uses a virtual time function defined with respect to GPS, which is accurate but too complex to implement, and selects for transmission the packet with the smallest virtual time (simpler algorithms like *Self Clocked Fair Queuing* (SCFQ) [10] have been proposed to compute the virtual time function). In WFQ the weights are fixed and depend on pre-defined service shares among the flows.

The main difference between LQF and OCF is that the former favours large queues, which can lead to permanent starvation of short queues, while the latter gives priority to cells with large waiting times.

LTRB schedules messages from input buffers; once a message has completed service, the next message is picked from the buffer that would overflow first, under the hypothesis that the input rate was the maximum allowed and no bandwidth was allocated to that flow. This policy is an improvement of LQF because it presents advantages of storage requirements per channel that do not increase with the number of incoming channels when compared to the logarithmic growth under LQF and FIFO.

III. RATIONALE FOR THE METHOD

A. Assumptions and Goals

Edge Devices have to process individual flows and map them into aggregate flows that share a common bandwidth (either fixed or variable depending on the service category) and will be given a global QoS guarantee.

We consider a particular class of flows that are shaped (and policed) according to TB parameters.

The TB model has been extensively studied in the literature as a shaping and policing mechanism, but usually this is not related to the resource allocation and scheduling policies. The main goal of this paper is to exploit this relationship, by using the TB parameters as a basis for resource allocation and scheduling decisions.

The method we propose is quite general and therefore may be used when either a VBR or an ABR service is chosen to transport the aggregate of TB flows. For simplicity we assume that traffic is transported in cells.

Conforming cells at the output of each TB are then subject to scheduling for sharing the available bandwidth. In the first place this bandwidth depends on the contracted service and the performance requirements; the achievable statistical gain depends, among other factors, on the number of multiplexed flows, the burstiness of the individual flows and the degree of statistical independence between flows. Variable rate channels, with some minimum guarantees, provide a higher degree of flexibility and may be used to achieve hard or soft performance bounds. However, whatever method is used to reserve and allocate bandwidth, short term scheduling (bandwidth sharing) of the individual flows is still required.

The proposed method attempts to put these aspects - TB policing, instantaneous bandwidth allocation and flow scheduling - into a common framework.

B. Model

According to the previous assumptions, we consider a network with the topology represented in Fig. 1.

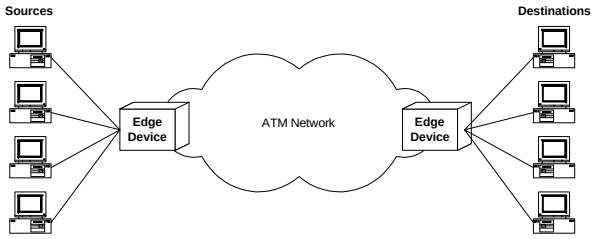

Fig. 1 The network configuration

The method is applied to an Edge Device that handles a number of input controlled flows. Each input port consists of a TB with an (optional) upstream buffer (to delay/shape non conforming cells) and a downstream buffer to hold cells that are waiting to be scheduled. All the scheduled traffic is aggregated into a common output buffer as illustrated in Fig. 2. A single ATM connection (ABR or VBR) is shared by all flows.

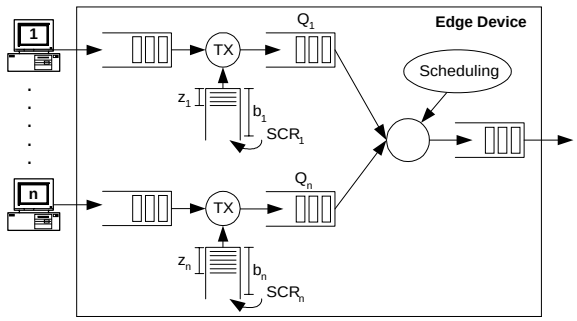

Fig. 2 The Edge Device architecture

The scheduler selects the next cell to be transmitted from one of the TB downstream buffers (queues) according to some rule based on individual TB states as well as the overall state. In the next Section, one such rule is presented to illustrate the method, but other scheduling algorithms are possible.

IV. SCHEDULING METHOD

A. Description of the Method

A TB compliant flow is subject to known bounds. For the purpose of this paper this can be easily illustrated by considering periodic ON-OFF sources. They are frequently used in the literature since they possibly represent the most demanding pattern among all (infinite) compliant flows. Assume that a source transmits during T_{ON}

$$T_{ON} = b /(PCR - SCR) \quad (3)$$

a number of cells equal to MBS at PCR and then stays inactive during an interval T_{OFF}

$$T_{OFF} = b / SCR \quad (4)$$

such that a compliant MBS can be accepted again, while attaining an average rate SCR during the whole period (T_{OFF} is the minimum time to replenish the TB).

Call *z* the number of available tokens on the TB and consider initially $z = b$ (bucket full). Let us assume that the flow is allocated a fixed bandwidth equal to SCR and that *Q* is the number of cells queued downstream the TB.

At $t = 0$, $z = b$ and $Q = 0$; then *z* decreases and *Q* increases and at $t = T_{ON}$, $z = 0$ and $Q = b$; then the queue is drained, still at SCR, and at $t = T_{ON} + T_{OFF}$ the initial conditions are reestablished. At any instant $z + Q = b$ holds.

However, if the channel bandwidth is $k * SCR$ $(k > 1)$, *Q* increases at a smaller rate and the queue is drained more quickly (now $z + Q < b$). Let us call $x = b - (z + Q)$; in this case $x > 0$ (while $x = 0$, in the first case), which means that the flow has been allocated a bandwidth larger than *SCR*.

We can now give an interpretation of the above parameters:

- *z* represents the number of available tokens (credits) in the TB; a large *z* means that a long, yet conforming burst, may still occur, while a small *z* means that only short bursts are expected;
- $b - z = Q + x$ represents the number of used tokens and therefore the number of cells submitted for scheduling, thus requiring transmission tokens (scheduling slots);
- *x* represents the number of transmission tokens that were allocated on excess of the (average) *SCR* allocation;
- *Q* represents the number of transmission tokens not yet allocated (queued cells).

The parameters (*z*, *x*, *Q*) are related to (*PCR*, *SCR*, *MBS*) and to the actual input and output rates and are quite useful in characterizing the state of a flow inside an Edge Device.

These parameters are represented in Fig. 3, for a particular instant in time, considering constant input and output rates (R_{IN}, R_{OUT}), such that PCR > R_{IN} > R_{OUT} > SCR. The ON-OFF source is represented by OBD, while OBC represents the upper bound for a compliant TB flow and OD represents a constant SCR allocation.

When multiplexing a number of TB flows, and assuming they are statistically independent, the bandwidth may be traded off between flows, by means of an adequate scheduling policy, according to their states.

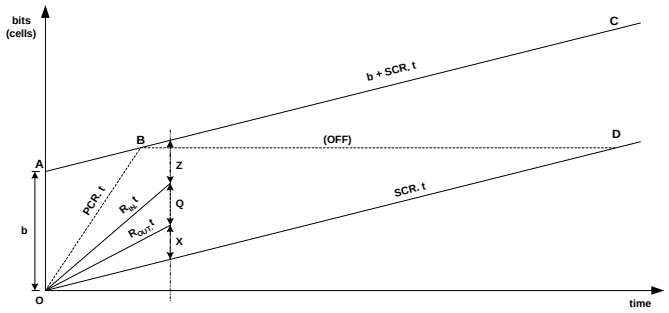

Fig. 3 Token Bucket Parameters

As an example, let us consider a number of similar periodic ON-OFF sources; (z_i, x_i, Q_i) represents the sate of each flow, while (*z*, *x*, *Q*) is the overall state.

If the ON periods of the sources do not overlap, *z* is constant (aggregate input rate constant). This means that the allocation of a constant bandwidth ($SCR = \Sigma\ SCR_i$) would allow scheduling the flows without any queuing; a *CBR* channel would be feasible and the best possible performance would still be achieved.

On the contrary, if the flows are strongly correlated (ON periods coincident), *z* would have the maximum possible excursion (between *0* and $b = \Sigma\ b_i$). In this case allocating a fixed bandwidth would produce the worst performance (similar to the case of a single flow with constant SCR_i allocation). Therefore this suggests the exploitation (if possible) of a VBR channel, with the highest possible allocation during the ON periods and smaller allocations during the OFF periods.

The first conclusion is that the short-term peak-to-peak excursion of *z* provides valuable information concerning a possible global bandwidth allocation strategy (that is, sharing of bandwidth with other traffic classes).

But the problem of scheduling still remains. Unlike the scheduling algorithms described in Section II, our proposal is to use the TB states as the criterion for scheduling.

In the first place, TB policing ensures that the flows are compliant. On the other hand it is expected that resources have been reserved (in a deterministic or statistical manner) according to declared traffic parameters and performance requirements; it is even possible that they have been tuned to the instantaneous requirements, as suggested. Then it remains to fairly share the available bandwidth (whatever its value) between flows according to their actual state (how large is the queue, how much bandwidth above the average has been used, how long is the maximum expected burst) and not to negotiated average or peak rate requirements.

Of course the TB state information can be used as an input to different criteria and this will be further exploited. Just for demonstrating the usefulness of the method, we

have considered due to its simplicity, the LTRB algorithm, with additional constraints that apply to TB flows.

B. Example of a Scheduling Algorithm

The LTRB algorithm has been adapted to the case of TB regulated flows. The basic idea was retained: the next flow to be scheduled is the one whose queue would saturate first under the hypothetical condition that the input rate was the maximum and the output rate the minimum possible. According to the previous analysis we consider that saturation is reached when $Q_i = b_i$ (since this would be the worst case under a fixed SCR_i allocation). Moreover since a compliant TB source cannot transmit at PCR_i permanently, we introduced the restriction that once z_i would reach *0*, the source would continue transmitting at SCR_i (and thus z_i would stay at *0*); similarly, before saturation is reached, we impose the output rate to be zero (the minimum possible), but afterwards it should have a value such that the saturation level would not be exceeded.

Considering the current state of a flow (z_i, x_i, Q_i) and imposing the above conditions, there are two cases of interest, depending on which condition ($z_i = 0$ or $x_i = 0$) is reached first.

In the first case saturation is reached after (Δt^1_i), such that

$$\Delta t^1_i \leftarrow x_i / SCR_i \quad (5)$$

(this corresponds to the time to reach $x_i = 0$).

In the second case the time to reach saturation is (Δt^2_i), such that

$$\Delta t^2_i \leftarrow (b_i - Q_i) / PCR_i \quad (6)$$

(when saturation is reached, $z_i > 0$ and $x_i = - z_i$).

Therefore the time Δt_i necessary to saturate queue *i* is the maximum of Δt^1_i and Δt^2_i; according to the algorithm, the flow to be scheduled is the one which has the minimum saturation time. The algorithm is the following:

```
/* Initialization */
01.  Δt = 0
02.  MinTime = 99999

/* Compute the source that will be served */
01.  for ∀i (queue not empty)
02.       X_i ← b_i - (Q_i + Z_i)
03.       Δt^1_i ← X_i / SCR_i
04.       Δt^2_i ←(b_i -Q_i) / PCR_i
05.       Δt ←MAX (Δt^1_i, Δt^2_i)
06.       if (MinTime > Δt)
07.            MinTime ←Δt
08.            SchSource ←i

/* Scheduling a cell from SchSource queue */
01.  Q_ATM(t) ←Q_SchSource(t)
```

V. PERFORMANCE ANALYSIS

The described algorithm has been evaluated by means of simulation, in order to assess the merits of scheduling traffic flows based on TB parameters. A number of scenarios were considered and two parameters analysed: queue occupancy and cell delay.

A. Traffic Model and Performance Parameters

In the simulation, ON-OFF sources have been used, as explained, and the ON and OFF periods selected so that flows were compliant with TB. The following parameters were used:

- PCR = 4 Mbit/s
- SCR = 1 Mbit/s
- MBS = 200 cells (b = 150)

Hence, T_{ON} = 20 ms and T_{OFF} = 60 ms (approximately).

Four sources with similar parameters were simulated; the ratio PCR/SCR was also set to four, so that by staggering the start of the ON periods of the sources (thus changing their relative phases) various degrees of correlation between them could be achieved - from totally overlapping to totally disjoint sources.

The extreme cases are trivial and the expected results were confirmed by simulation - in the first case the four sources equally shared the available bandwidth while, in the latter, each source was allocated its PCR, since there was no competition among sources.

Three non trivial cases were considered, by staggering the sources as shown in Table I.

Table I
Starting Times of ON periods

Scenarios	Sources			
	S1 (ms)	S2 (ms)	S3 (ms)	S4 (ms)
1	0	5	10	15
2	0	10	20	30
3	0	15	30	45

In the simulations described in this paper only a fixed bandwidth channel was considered with a rate slightly above (10%) the overall average rate of the aggregate flow; further simulations will address the case of variable bit rate channels, introducing higher and lower priority traffic classes.

B. Simulation Results

In Fig. 4-6 the queue occupancies and the cell delays are shown for the three scenarios under study; the cell delay is represented as a function of the departure time.

As a term of comparison, if each ON-OFF flow were allocated a fixed bandwidth equal to its SCR, the maximum cell delay would be in the order of 60 ms (T_{OFF}).

The figures put in evidence some interesting features of the scheduling algorithm.

In the first place the algorithm performs quite well in controlling the maximum buffer occupancy. The flows with lower indices (earlier starting times) get initially a larger share of the bandwidth and therefore their maximum queue

occupancy is smaller. However, the draining of their queues is slower, which manifests in the larger delays they suffer (the sources with higher indices get a comparatively larger share of the bandwidth at the end of their activity periods).

The distribution of cell delays is not quite fair, but this is a property of the scheduling algorithm and not of the proposed method itself. Other scheduling algorithms that may overcome this limitation are still being investigated.

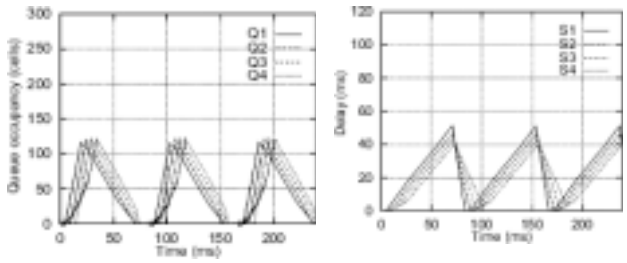

Fig. 4 Queue occupancy and cell delay for scenario 1.

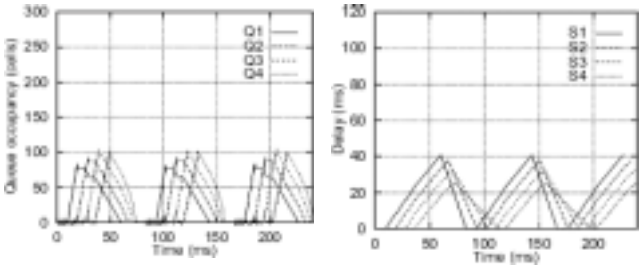

Fig. 5 Queue occupancy and cell delay for scenario 2.

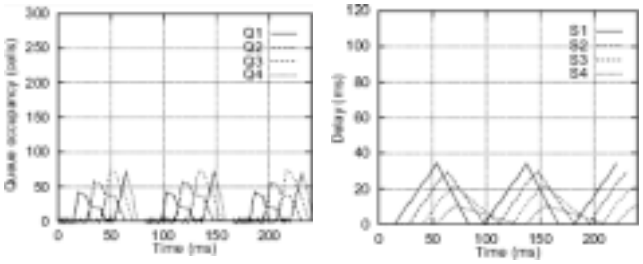

Fig. 6 Queue occupancy and cell delay for scenario 3.

These results seem very promising and although the LTRB scheduling algorithm presents some intrinsic limitations, it provides some useful insight into the nature of the proposed method and therefore provides some clues for the investigation of other algorithms.

VI. CONCLUSIONS

This paper presents some preliminary results of a new method that can be applied to the scheduling of TB flows and to the dynamic allocation of bandwidth to aggregates of such flows. The method will be further applied to other scheduling algorithms, by exploiting various criteria for distributing the instantaneous bandwidth among flows based on the individual as well as the overall TB states.

The method provides information about the degree of correlation between flows and therefore on the short-term bandwidth requirements, ranging from fairly constant to highly variable bit rates; this may be easily used for bandwidth renegotiation purposes.

The method can be exploited with variable bit rate channels, provided that a minimum bandwidth is guaranteed; this is the case of ABR channels. In this case it gives a criterion to specify the short term PCR, while the ACR indicated by the network can be used not only to assist scheduling but also to provide feedback to sources to adapt their TB parameters (an example could be video sources adapting their coding parameters). This issue will be further investigated.

VII. REFERENCES

[1] ATM Forum, "ATM Traffic Management Specification", version 4.1", April 1999.

[2] M. Sidi, W.-Z. Liu, I. Cidon and I. Gopal, "Congestion Control Through Input-rate Regulation", *IEEE Trans. Communications*, vol. 41, March 1993, pp. 471-477.

[3] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: An Overview", RFC 1633, June 1994.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[5] S. Rosenberg, M. Assaoui, K. Galway and N. Giroux, "Functionality at the Edge: Designing Scalable Multiservice ATM Networks", *IEEE Communication Magazine* , May 1998, pp. 88-99.

[6] A. Demers, S. Keshav and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", *Journal of Internetworking Research and Experience*, October 1990, pp. 3-26.

[7] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving 100% Throughput in an Input Queue Switch", *Conference Proceedings IEEE INFOCOM'96*, San Francisco, CA, USA, March 1996, pp. 296-302.

[8] A. Birman, H. R. Gail, S. L. Hantler, Z. Rosberg and M. Sidi, "An Optimal Service Policy for Buffer Systems", *IBM Research Report*, 1995.

[9] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control - the Single Node Case", IEEE/ACM Trans. Networking, June 1993, pp. 344-357.

[10] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Applications", *Conference Proceedings IEEE INFOCOM'96*, San Francisco, CA, USA, March 1996.