ISSN: 2581-9674
Article ID: IJDCSE05202005
International Journal for Development of Computer Science and Engineering (IJDCSE)
Volume-2, Issue-3, May-2020, PP.29-36
www.ijdcse.org
Received: 5-03-2020, Accepted: 07-04-2020, Published: 08-05-2020 .

Design and analysis of GCRA traffic shaper for asynchronous transfer mode for preventing congestion control

Er.L .K. Lakshminarayanan.(Research Scholar)

Abstract

This study deals with the enforcement of source parameters in ATM networks An algorithm for monitoring the cell rate of a connection is the so called generic cell rate algorithm At the moment only peak cell rate policing is standardized but there are discussions in the standard commissions to introduce a second source parameter the sustainable cell rate in conjunction with a burst tolerance. The Project flow chart algorithm is multicast service. It is very simple being LAN/WAN broadcasting tool. The LAN/WAN links are often private Lines, unlike submarine over network. A private network has the advantage of being managed and by few people so to avoid many problems about the property and origin of LAN/ WAN has been investigated in the literature for some use. The fundamental philosophy behind the internet is expressed by scalability argument No protocol, mechanism or service should be introduced in to the internet if it does not scale well. A key corollary to the scalability argument is the end to end argument to maintain scalability algorithmic complexity should be pushed to the edges of the network to whenever possible Perhaps the best example of the internet philosophy the TCP congestion control which is implemented primarily to algorithms operating at end systems unfortunately TCP congestion control also illustrates some of the shortcomings the end to end argument As a result of its strict adherence to end and congestion control.

Proposed System

- Buffering of packets is carried out in the edge routers rather than in the core routers.
- NBP is to compare, at the borders of a network i.e. the rate at which each packet from each application flow are entering and leaving the network.
- In addition to NBP algorithm for prevention of congestion collapse. The Feedback control Algorithm and Rate Control Algorithm acts together to finish it thereby introducing Traffic shaper concept(GCRA)
- To preventing the Congestion Control through the communication over network we are introducing Intelligent Packet Filtering using GCRA Concept.
- NBP's prevention of congestion collapse comes at the expense of some additional network complexity, since routers at the border of the network are expected to monitor and control the rates of individual flows.

Problem Methodology

System Flow diagram are directed graphs in which nodes specify processing activities and arc specify data item transmitted between processing nodes .Data Flow diagrams represent the system between individual items

The Generic Cell Rate Algorithm

The generic cell rate algorithm (GCRA) is a leaky bucket-type scheduling algorithm for the network scheduler that is used in Asynchronous Transfer Mode(ATM) networks. It is used to measure the timing of cells on virtual channels (VCs) and or Virtual Paths (VPs) against bandwidth and jitter limits contained in a traffic contract for the VC or VP to which the cells belong. Cells that do not conform to the limits given by the traffic contract may then be re-timed (delayed) in traffic shaping, or may be dropped (discarded) or reduced in priority (demoted) in traffic policing. Nonconforming cells that are reduced in priority may then be dropped, in preference to higher priority cells, by downstream components in the network that are experiencing congestion. Alternatively they may reach their destination (VC or VP termination) if there is enough capacity for them, despite them being excess cells as far as the contract is concerned: see priority control.

The GCRA is given as the reference for checking the traffic on connections in the network, i.e. usage/network parameter control (UPC/NPC) at user–network interfaces (UNI) or inter-network interfaces or network - network interfaces (INI/NNI) .It is also given as the reference for the timing of cells transmitted (ATM PDU Data Requests) onto an ATM network by a network interface card (NIC) in a host, i.e. on the user side of the UNI . This ensures that cells are not then discarded by UPC/NCP in the network, i.e. on the network side of the UNI. However, as the GCRA is only given as a reference, the network providers and users may use any other algorithm that gives the same result. The GCRA is reference algorithm for a cell rate which determines if a cell is conforming. [3]The GCRA is a relatively simple algorithm given as a flowchart in the following Figure 2.

Figure 2

The action of a leaky bucket can be dimensioned with two parameters: the drain rate of the bucket (I) and the height of the bucket (L).The greater the drain rate, the faster the cells pour out of the bucket. The greater the height of the bucket, the more cells the bucket can buffer. If the cells are pouring too quickly into the bucket, the bucket will overflow and cells will be lost. The algorithm defines a finite capacity bucket that drains at a continuous rate of one per time unit and whose content is increased by I for each conformed cell. The total capacity of the bucket is L. After the arrival of the kth cell at t_a (k), the algorithm checks to see if the bucket has over flown. If so, the cell is discarded. If not, the bucket is incremented. The amount of the increment depends on whether the bucket was fully drained between cell arrivals. The GCRA is a reference algorithm for determining the cell rate conformance. Earlier we have introduced different traffic descriptor parameters such as PCR, SCR and BT. how we use the GCRA with these parameters figure 3.

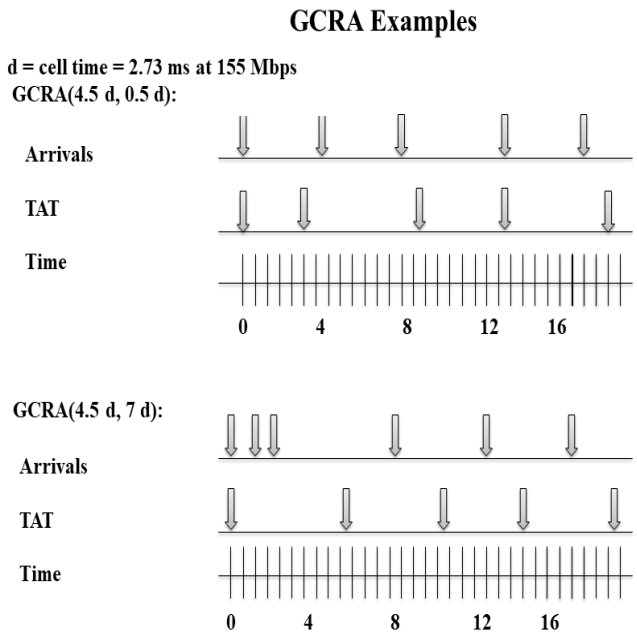

Figure -3

Leaky Bucket Algorithm

The description in terms of the leaky bucket algorithm may be the easier of the two to understand from a conceptual perspective, as it is based on a simple analogy of a bucket with a leak: see figure 1 on the leaky bucket page. However, there has been confusion in the literature over the application of the leaky bucket analogy to produce an algorithm, which has crossed over to the GCRA. The GCRA should be considered as a version of the leaky bucket as a meter rather than the leaky bucket as a queue. However, while there are possible advantages in understanding this leaky bucket description, it does not necessarily result in the best (fastest) code if implemented directly. This is evidenced by the relative number of actions to be performed in the flow diagrams for the two descriptions figure 4

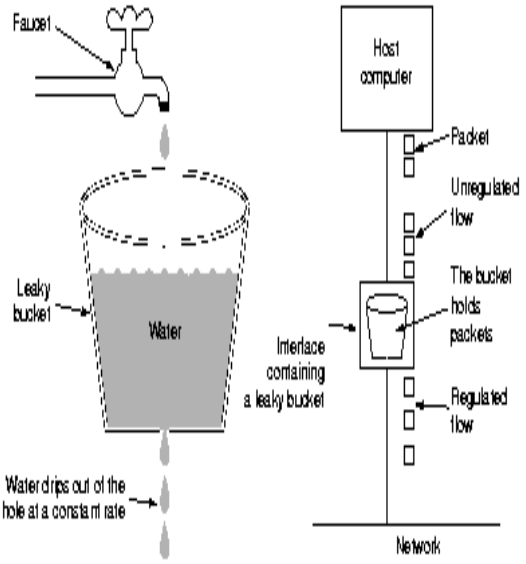

Figure 4

The description in terms of the continuous state leaky bucket algorithm is given by the ITU-T as follows: "The continuous-state leaky bucket can be viewed as a finite capacity bucket whose real-valued content drains out at a continuous rate of 1 unit of content per time unit and whose content is increased by the increment T for each conforming cell... If at a cell arrival the content of the bucket is less than or equal to the limit value τ, then the cell is conforming; otherwise, the cell is non-conforming. The capacity of the bucket (the upper bound of the counter) is $(T + \tau)$" [2] It is worth noting that because the leak is one unit of content per unit time, the increment for each cell T and the limit value τ are in units of time.

Considering the flow diagram of the continuous state leaky bucket algorithm, in which T is the emission interval and τ is the limit value: What happens when a cell arrives is that the state of the bucket is calculated from its state when the last conforming cell arrived, X, and how much has leaked out in the interval, ta – LCT. This current bucket value is then stored in X' and compared with the limit value τ. If the value in X' is not greater than τ, the cell did not arrive too early and so conforms to the contract parameters; if the value in X' is greater than τ, then it does not conform. If it conforms then, if it conforms because it was late, i.e. the bucket empty ($X' <= 0$), X is set to T; if it was early, but not too early, ($\tau >= X' > 0$), X is set to $X' + \tau$.Thus the flow diagram mimics the leaky bucket analogy (used as a meter) directly, with X and X' acting as the analogue of the bucket.

Token Bucket

The token bucket algorithm is based on an analogy of a fixed capacity bucket into which tokens, normally representing a unit of bytes or a single packet of predetermined size, are added at a fixed rate. When a packet is to be checked for conformance to the defined limits, the bucket is inspected to see if it contains sufficient tokens at that time. If so, the appropriate

number of tokens, e.g. equivalent to the length of the packet in bytes, are removed ("cashed in"), and the packet is passed, e.g., for transmission. The packet does not conform if there are insufficient tokens in the bucket, and the contents of the bucket are not changed. Non-conformant packets can be treated in various ways:

- They may be dropped.
- They may be enquired for subsequent transmission when sufficient tokens have accumulated in the bucket.
- They may be transmitted, but marked as being non-conformant, possibly to be dropped subsequently if the network is overloaded.

A conforming flow can thus contain traffic with an average rate up to the rate at which tokens are added to the bucket, and have a burstiness determined by the depth of the bucket. This burstiness may be expressed in terms of either a jitter tolerance, i.e. how much sooner a packet might conform (e.g. arrive or be transmitted) than would be expected from the limit on the average rate, or a burst tolerance or maximum burst size, i.e. how much more than the average level of traffic might conform in some finite period.

Comparison with the Token Bucket

The GCRA, unlike implementations of the token bucket algorithm, does not simulate the process of updating the bucket (the leak or adding tokens regularly). Rather, each time a cell arrives it calculates the amount by which the bucket will have leaked since its level was last calculated or when the bucket will next empty (=*TAT*). This is essentially replacing the leak process with a (real time) clock, which most hardware implementations are likely to already have.

This replacement of the process with an RTC is possible because ATM cells have a fixed length (53 bytes), thus *T* is always a constant, and the calculation of the new bucket level (or of *TAT*) does not involve any multiplication or division. As a result, the calculation can be done quickly in software, and while more actions are taken when a cell arrives than are taken by the token bucket, in terms of the load on a processor performing the task, the lack of a separate update process more than compensates for this. Moreover, because there is no simulation of the bucket update, there is no processor load at all when the connection is quiescent.

However, if the GCRA were to be used to limit to a bandwidth, rather than a packet/frame rate, in a protocol with variable length packets (Link Layer PDUs), it would involve multiplication: basically the value added to the bucket (or to TAT) for each conforming packet would have to be proportionate to the packet length: whereas, with the GCRA as described, the water in the bucket has units of time, for variable length packets it would have to have units that are the product of packet length and time. Hence, applying the GCRA to limit the bandwidth of variable length packets without access to a fast, hardware multiplier (as in an FPGA) may not be practical. However, it can always be used to limit the packet or cell rate, as long as their lengths are ignored.[1]

Dual Leaky Bucket Controller

Multiple implementations of the GCRA can be applied concurrently to a VC or a VP, in a dual leaky bucket traffic policing or traffic shaping function, e.g. applied to a Variable Bit Rate (VBR) VC. This can limit ATM cells on this VBR VC to a Sustained Cell Rate (SCR) and a Maximum Burst Size (MBS). At the same time, the dual leaky bucket traffic policing function can limit the rate of cells in the bursts to a Peak Cell Rate (PCR) and a maximum Cell Delay Variation tolerance (CDVt).

This may be best understood where the transmission on an VBR VC is in the form of fixed length messages (CPCS-PDUs), which are transmitted with some fixed interval or the Inter Message Time (IMT) and take a number of cells, MBS, to carry them; however, the description of VBR traffic and the use of the dual leaky bucket are not restricted to such situations. In this case, the average cell rate over the interval of IMT is the SCR (=MBS/IMT). The individual messages can be transmitted at a PCR, which can be any value between the bandwidth for the physical link ($1/\delta$) and the SCR. This allows the message to be transmitted in a period that is smaller than the message interval IMT, with gaps between instances of the message.

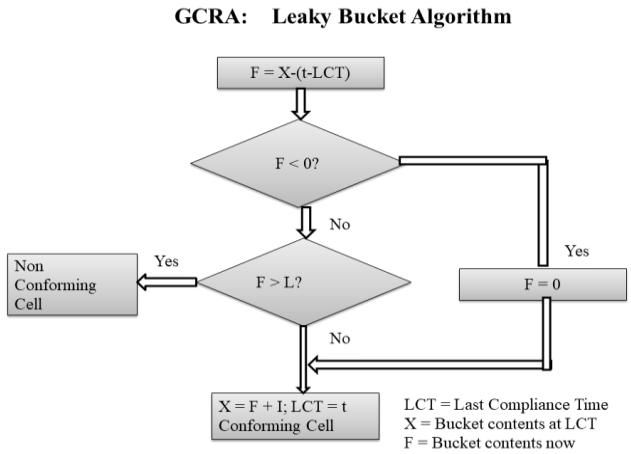

Figure -5

In the dual leaky bucket, one bucket is applied to the traffic with an emission interval of 1/SCR and a limit value TSCR that gives an MBS that is the number of cells in the message: see leaky bucket# Maximum Burst Size. The second bucket has an emission interval of 1/PCR and a limit value TSCR that allows for the CDV up to that point in the path of the connection: see leaky bucket Delay Variation Tolerance. Cells are then allowed through at the PCR, with jitter of TSCR, up to a maximum number of MBS cells. The next burst of MBS cells will then be allowed through starting MBS x 1/SCR after the first.

If the cells arrive in a burst at a rate higher than 1/PCR (MBS cells arrive in less than (MBS - 1)/PCR - TSCR), or more than MBS cells arrive at the PCR, or bursts of MBS cells arrive closer than IMT apart, the dual leaky bucket will detect this and delay (shaping) or drop or

de-prioritize (policing) enough cells to make the connection conform. Figure -5 shows the reference algorithm for SCR and PCR control for both Cell Loss Priority (CLP) values 1 (low) and 0 (high) cell flows, i.e. where the cells with both priority values are treated the same. [4]

Conclusion

In this paper we have presented an analysis method to determine the exact cell rejection probability of the GCRA a novel congestion avoidance mechanism for the Internet called Network Border Patrol. Unlike existing Internet congestion control approaches, which rely solely on end-to-end control, NBP is able to prevent congestion collapse from undelivered packets. It does this by ensuring at the border of the network that each flow's packets do not enter the network faster than they are able to leave it. NBP requires no modifications to core routers or to end systems. Only edge routers are enhanced so that they can perform the requisite per-flow monitoring, per-flow rate control and feedback exchange operations.

Reference

1. Jump up to atm forumthe user network interface (uni), v. 3.1, isbn 0-13-393828-x, prentice hall ptr, 1995.
2. Jump up to itu-t, traffic control and congestion control in b isdn, recommendation i.371, international telecommunication union, 2004,
3. M_ ritter january. 1994. Analysis of the generic cell rate algorithm monitoring on_off_traffc ,http://www-info3.informatik.uni-wuerzburg.de/tr/tr077institute of computer science, university of wurzburg
4. Itu-t, traffic control and congestion control in b isdn, recommendation i.371, international telecommunication union, 2004,

Address:

T. LAKSHMINARAYANAN*(Research Scholar)
Department of Computer Science
Periyar University College of arts and science
Mettur Dam-636401
lakshmitvr@rediffmail.com

DR. R. UMARANI**
Associate Professor
Department of Computer Science
Sri Sarada College for Women
Salem -07