

Adaptive Token Allocation Algorithm For Enhancing The Fairness Among Long Live TCP Flows In Diffserv

S.Sudha*, N. Ammasaigounden**

*sudha@nitt.edu, **ammas@nitt.edu

National Institute of Technology, Tiruchirappalli, Tamil Nadu

Abstract: In this paper, an attempt has been made to design an algorithm that provides optimal allocation of tokens to the individual long live TCP flows in Differentiated Services (DiffServ) networks. The optimality in token allocation is achieved using an optimization technique that employs Genetic Algorithm (GA). The trial performance of the proposed algorithm has been evaluated through C++ simulation. It has been observed from the simulation studies that the token allocation varies with respect to the incoming rate of the flows. The formulation of the objective function and the application of GA for the optimization process have been explained. The simulation results pertaining to the token allocation for two different cases are also furnished.

Keywords: Token Allocation, Packet Marking, DiffServ networks, optimization, Genetic Algorithm.

I. INTRODUCTION

Diffserv defines the architecture for implementing scalable service differentiation in the Internet [1]. These networks can provide QoS over and above the current *best-effort* service. The most prevalent forwarding mechanisms standardized by the Internet Engineering Task Force (IETF) are *Expedited Forwarding Per-Hop Behavior (EF PHB)* [2] and *Assured Forwarding Per-Hop Behavior (AF PHB)* [3]. The former is intended to support traffic flows requiring a short delay and the latter is intended to assure a minimum level of throughput.

To assure this minimum throughput, which is referred to as the *target rate* or *committed information rate (CIR)*, AF PHB introduces two components; a packet marking mechanism administrated by *profile meters* or *traffic conditioners* at edge routers and a queue management mechanism at core routers. The most widely deployed queue management algorithm is the RED-based algorithm (RED with In/Out (RIO)) [4]. The traffic sources are aggregated to several classes with different QoS guarantees. The *traffic conditioner* is an important element of DiffServ, which consists of the meter, marker and shaper or dropper [7]. The most prevalent profile meters are the Token Bucket (TB) based marker [4]-[6] and the average rate estimator based (Time Sliding Window (TSW)) marker [8]-[11].

The meter is designed to measure the passing traffic against the traffic profile and pass the conforming level of the packet to the marker. The marker determines the color of the packet to be green, yellow, or red corresponding to different conforming levels according to the metering result, and mark the packet with an appropriate DiffServ Code Point (DSCP)

[7]. The traffic conditioner would shape or discard the packet to meet the traffic profile by the shaper or dropper, respectively. When packets exit from the traffic conditioner, the marking result will significantly affect the forwarding behavior of micro-flows.

The micro-flows that require the similar QoS level would be assigned to the same class. However, the conventional traffic conditioner works on the aggregate traffic of the class and never discriminates from one micro-flow to others. The unfair marking among micro-flows appears in conventional marking algorithms, especially in the models that apply token bucket mechanism.

Feroz et al. have proposed a TCP friendly marker to minimize the spacing between the subsequent 'IN' packets to overcome the bursty nature of the traffic, which is the main cause for the packet losses in TCP. The small window flows are protected by an equal allocation of tokens and the long live flows are allocated using max-min-fair allocation algorithm [12]. The simulation results show a slight improvement in the average goodput. By using max-min-fair allocation algorithm, the tokens are almost equally allocated to all flows, by which fairness does not exist.

In this paper, we focus on the unfair marking among the flows in an aggregate and propose a new approach using the optimization technique employing genetic algorithm to distribute the available tokens in a fair manner to the individual flows in a DiffServ network. This algorithm has been formulated based on the Token Bucket policy.

II. CONVENTIONAL TRAFFIC CONDITIONER

There are many reasons for the unfair marking in the conventional traffic conditioner. In this paper we focus only on some accounts that are obvious and effective largely are listed as follows:

A. Periodic traffic: As described in [13], many traffic sources are highly periodic, such as real-time audio/video or traffic with window flow control protocol that has periodic cycle equal to the round trip time like TCP bulk data transfer. When the micro-flows of periodic sources enter the token-bucket-based marker, it would cause unfair marking and the phenomenon is also called phase effect [14].

B. Coexistence of adaptive traffic and non-adaptive traffic:

Adaptive traffic (e.g. TCP) is possible to reduce the sending rate if congestion happens, but non-adaptive like UDP traffic source changes nothing under the congestion situation, therefore, the congestion brings lower sending rate to adaptive traffic but no effect in non-adaptive traffic. The unfair bandwidth allocation happens if we do not protect the adaptive traffic.

C. Traffic of greedy user: Because the conventional traffic conditioner polices the sending rate of aggregate traffic, a user is usually allocated the bandwidth proportional to the sending rate. When a greedy user increases the sending rate, he/she would get more bandwidth. This leads to the unfair marking.

III TOKEN BUCKET MODEL

It is well known that for a long-lived TCP flow under the assumption of constant round-trip time (RTT) and low to medium periodic packet losses, the TCP congestion control window (CWND) follows a saw-tooth like pattern. Based on this assumption, Mathis et al. [15] described the behavior of a single TCP flow by

$$BW = \frac{(MSS)c}{(RTT)\sqrt{p}}$$

where BW :achieved bandwidth, in bytes per second (byte/s). This represents the average bandwidth achieved by an aggregate.

MSS : the maximum segment size in bytes

RTT : the round-trip time in milliseconds.

p : the probability of packet loss in the network

c : a constant

Under the assumption that packet losses are periodic, and the destination acknowledges every packet, c is equal to $\sqrt{3/2}$.

This model captures only the congestion avoidance phase of the TCP congestion control algorithm [16]. From [15] for an aggregate with n flows, the parameters for the individual flows are

$$CWS = \frac{(CIR)(RTT)}{(MSS)n} \quad (2)$$

$$B_p = \frac{B}{n(MSS)}$$

where CIR : committed information rate, in bytes per second, is the desired bandwidth of an aggregate. It is also used as the token arrival rate of the token bucket marker.

CWS: committed window size for a TCP flow, in packets.

B : token bucket size measured in bytes.

B_p:token bucket size measured in packets.

This unfair marking among micro-flows in token bucket mechanism, wherein a greedy user might get more bandwidth than others will cause unfairness. To overcome such

unfairness, it is required that each and every flow of an aggregate get an optimal bandwidth.

In paper [12], it is proved that minimizing packet losses by providing minimal spacing between the 'IN' packets, reduce timeouts. This reduction in timeouts improve goodput. With a similar notion, to handle fairness an objective function has been formulated to minimize the packet losses of the individual flows and thus achieve a fair allocation of tokens for these individual flows. The next section will brief the design principles of the proposed work.

IV. PROPOSED WORK

To design an algorithm for optimal allocation, the state information of the flows like (i) the number of flows in an aggregate (ii) the estimate of the sending rate of the individual flows and the token bucket parameters such as (i) the token bucket size are required. Due to the above fact, the per-flow state has to be maintained.

Based on the per-flow statistics and the availability of the tokens in the token bucket at an interval 'T' msec, an optimization algorithm is proposed to obtain a token allocation vector for the flows. The tokens are to be optimally allocated with an objective to minimize the packet losses of the individual flows, thereby increasing the throughput of the flows. This requirement is met by using appropriate weighting factor in the formulation of the objective function which is evaluated by an optimization process. The output of the optimization process results in a token allocation vector X and the default spacing vector O of size 'n', where 'n' corresponds to the number of flows in an aggregate.

Hence, the objective function for the proposed work is formulated as :

$$F(x) = \sum_{i=1}^n w_i (p_i - x_i)$$

p_i = subject to the following constraints :

$$g_1 = \sum x_i \leq B_p ; \quad i = 1, 2, \dots, n.$$

where

g₁, g₂ are the constraints

T = the marking interval

estimate of the number of packets of flow 'i' in the interval T

x_i = number of tokens allocated to flow 'i' in T seconds

n = number of flows

B_p= Bucket size in packets at T seconds

w_i = weight factor for flow 'i', expressed as the fraction of the packet estimation of flow 'i' to the total packet estimation of 'n' flows.

$$\text{i.e. } w_i = p_i / \sum p_i$$

Using the objective function, an algorithm for token allocation has been designed and presented below.

Token Allocation Algorithm

Step I: Finding the total of the estimated packets
 The estimate of the No. of. packets for each flow i , find $\sum p_i$
 $\forall i$

Step II : Distribution of the tokens among the flows
 /* if tokens are less than the number of packets*/

If ($B_p < \sum P_i$; $i = 1,2,\dots,n$) Divide B_p among the flows
 optimally using the objective function

$$F(x) = \sum_{i=1}^n w_i (p_i - x_i)$$

/* vector X is obtained for the flows
 using GA subject to the constraints */
 $O_i = [p_i/x_i] - 1$;

/* if tokens are equal to the number of
 packets */

Else if ($B_p = \sum p_i$; $i = 1,2,\dots,n$)
 All packets are assigned tokens
 $x_i = p_i$
 $O_i = 0$

The token allocation vector X and the packet interleaving vector O (i.e. default spacing) for all the ‘n’ flows are obtained by the above algorithm. Based on the spacing vector O, the packets may be marked as either “IN” or “OUT” depending on the token allocation vector X of that particular flow. The packet marking done can be done using the one proposed by Feroz. et.al [12].

By providing such optimal spacing between the subsequent “IN” packets, the timeouts can be minimized thereby enhancing the performance of the marker. Further, the fairness among the flows may be achieved because of the optimal allocation of bandwidth. As a start, we have simulated this algorithm to study its static performance in C++ and it is planned to extend this in NS2.

V. OPTIMIZATION USING GA

The lower and the upper limits of the allocation vector X corresponding to the individual flows ‘i’ are initialized to 0(zero) and p_i respectively. Starting from the specification of the population size, number of generations, cross-over and mutation probability, the optimization process is initiated. In general, GA maximizes the given function whereas in the present case, the aim is to minimize the objective function (eqn.4). So, the function is written as $fitval = 1/(1+objval)$, where **objval** is the objective function value and **fitval** is the fitness function value. As a result of this optimization, the

token allocation and the default spacing for all the ‘n’ flows are obtained in a vector X and O respectively.

VI. NETWORK TOPOLOGY AND SIMULATION RESULTS

The following values have been chosen for simulation in the optimization process:

- Number of generations = 500 ; population size = 100;
- mutation probability = 0.05 ;
- cross-over probability = 0.7.

The simulation scenario has been taken for a network topology as shown in Fig.1, where the aggregate S0 is assumed to have ‘n’ simultaneous long-lived TCP flows and there is a bottleneck link, with destination D0. The simulation was carried out for 10 flows. The estimation of the number of packets corresponding to each flow ‘i’ was obtained using a random generator. The random generator follows a uniform distribution in the range 0 to 1. To have the effect of a real network, the estimates were scaled up to higher values. The simulations have been carried out for two cases namely i) when the network load is high and ii) when the network load is balanced. In the former case, the flows are protected by optimally allocating the available tokens and in the latter case all the flows are allocated equal to their requirement. The number of tokens was suitably initialized to simulate the above cases. The results of the simulation are plotted in the form of graphs shown in Fig.2. From the plots, it is observed that the token allocation varies with the estimation of the number of packets, thus providing an optimal allocation of tokens and thus providing fairness among the flows in an aggregate.

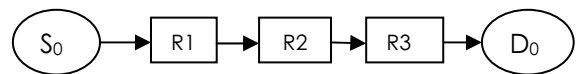
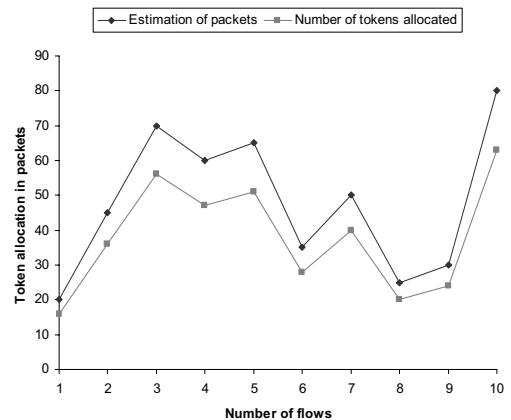
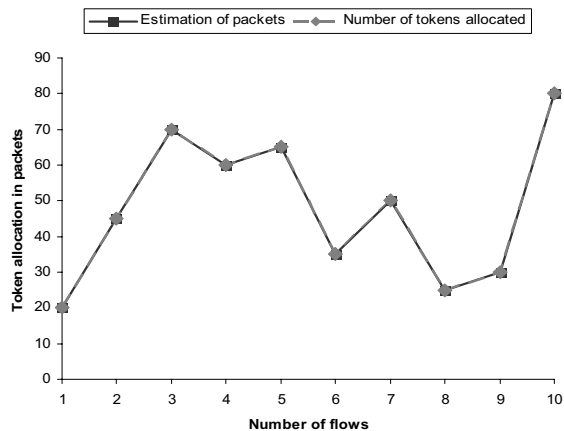


Fig. 1 Network topology



(a)



(b)
 Fig. 2 Plot of flow Vs token allocation for n=10
 (a) when network load is high
 (b) when network load is balanced

CONCLUSIONS AND FUTURE WORK

A new token allocation algorithm using GA technique has been developed for providing fairness among flows in a DiffServ network. The token allocation algorithm suggested provides an *optimal distribution* of tokens to the flows, when the network load is high or balanced. From the results, it is concluded that minimization of packet losses result in fair token allocation among the individual flows of an aggregate by judicious selection of the objective function. This algorithm has been designed to be invoked only at times of congestion. The default spacing so obtained is also optimal as it is expressed as the ratio of the optimal tokens allocated to the estimate of the packets corresponding to the flows. By this approach, the timeouts will be minimized as the bursty nature of the TCP traffic is overcome. This will definitely lead to an improved throughput in the network. As further work, it is aimed to evaluate the performance through simulations on NS (Network Simulator).

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *RFC 2475*, December 1998.
 [2] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," *RFC 2598*, June 1999.
 [3] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," *RFC 2597*, June 1999.

[4] D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," *IEEE/ACM Trans. on Networking*, vol. 6, no. 4, pp. 362–373, 1998.
 [5] Y. Chait, C. Hollot, V. Misra, D. Towsley, and H. Zhang, "Providing throughput differentiation for TCP flows using adaptive two color marking and multi-level AQM," *Proceedings of IEEE INFOCOM*, 2001.
 [6] W. Lin, R. Zheng, and J. Hou, "How to make assured service more assured," *Proceedings of IEEE ICNP*, 1999.
 [7] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers," *RFC 2474*, Dec. 1998.
 [8] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Adaptive packet marking for maintaining end-to-end throughput in a differentiated-services internet," *IEEE/ACM Trans. on Networking*, vol. 7, no. 5, pp. 685–697, 1999.
 [9] I. Yeom and A. L. N. Reddy, "Modeling TCP behavior in a differentiated services network," *IEEE/ACM Trans. on Networking*, vol. 9, no. 1, pp. 31–46, 2001.
 [10] N. Malouch and Z. Liu, "Performance analysis of TCP with RIO routers," *Proceedings of IEEE Globecom*, Nov., 2002.
 [11] S. Sahu, P. Nain, C. Diot, V. Firoiu, and D. Towsley, "On achievable service differentiation with token bucket marking for TCP," *Proceedings of ACM SIGMETRICS*, pp. 23–33, 2000.
 [12] A. Feroz, A. Rao and S. Kalyanaraman, "A TCP friendly traffic Marker for IP Differentiated services," *Proc. of IWQoS'00*, pp. 138–147, 2000.
 [13] F. Wang, P. Mohapatra, S. Mukherjee, D. Bushmitch, "A random early demotion and promotion marker for assured services," *IEEE Journal on Selected Areas in Commun.*, vol. 18, no. 12, Dec. 2000.
 [14] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, no. 3, pp. 115–156, Sep. 1992.
 [15] Mathis, M., Semke, J., Mahdavi, J., and Ott, T.: 'The macroscopic behavior of the TCP congestion avoidance algorithm', *Comput. Commun. Rev.*, 1997, 27, (1), pp. 67–82
 [16] Jacobson, V., and Karels, M.J.: 'Congestion avoidance and control'. *Proc. SIGCOMM'88*, Stanford, CA, USA, August 1988, pp. 314–329.