FARED: Fast Adapting RED Gateways for TCP/IP Networks

Mohit P. Tahiliani[1], K.C. Shet[1], and T.G. Basavaraju[2]

[1] Department of Computer Science and Engineering
N.I.T.K., Surathkal, India
tahiliani.nitk@gmail.com
[2] Department of Computer Science and Engineering
Govt. S.K.S.J. Technological Institute, Bangalore, India

Abstract. Random Early Detection (RED) is a widely deployed active queue management mechanism to improve the performance of the network in terms of throughput and packet drop rate. The effectiveness of RED, however, highly depends on appropriate setting of its parameters. In this paper, we propose a Fast Adapting Random Early Detection (FARED) algorithm which efficiently varies the maximum drop probability to improve the overall performance of the network. Based on extensive simulations, we show that FARED algorithm reduces the packet drop rate and achieves better throughput than Adaptive RED (ARED) and Refined Adaptive RED (Re-ARED). Moreover, FARED algorithm does not introduce new parameters to improve the performance and hence can be deployed without any additional complexity.

Keywords: Active Queue Management (AQM), packet drop rate, throughput.

1 Introduction

Tremendous growth in the number of internet users has led to an exponential increase in the internet traffic, making it complicated to handle network congestion. TCP congestion control mechanisms are widely deployed and extensively used by a variety of internet applications. Since most of these mechanisms consider network as a *black box*, they are limited to *packet drops* as the only indication of congestion[1]. Such mechanisms are not well suited for applications such as telnet, web browsing, etc which are sensitive to packet drops. Moreover, traditional drop-tail gateways do not provide an early congestion notification. This leads to global synchronization, a phenomenon in which all senders sharing the bottleneck gateway reduce their sending rate at the same time, thereby under-utilizing the network resources.

Random Early Detection (RED) gateways[2] overcome the drawbacks of drop-tail gateways by avoiding global synchronization and offer fairness among several competing end hosts. However, the effectiveness of RED largely depends on appropriately setting atleast four parameters, namely: minimum threshold (min_{th}), maximum threshold (max_{th}), queue weight factor (w_q) for exponential weighted

P.S. Thilagam et al. (Eds.): ADCONS 2011, LNCS 7135, pp. 435–443, 2012.

moving average and maximum drop probability (max_p)[3][4][5][6][7]. Optimal values for these parameters differ for different scenarios and are dependent on several other factors such as number of flows passing through same bottleneck gateway[3][5], packet size[8], etc.

Adaptive RED (ARED)[6] addresses the parameter sensitivity of RED by dynamically varying max_p and automatically setting min_{th}, max_{th} and w_q parameters. ARED requires setting of only one parameter - *target queuing delay*, defined as the maximum amount of time a packet is delayed in the queue. Though ARED dynamically varies max_p, the adaptation is conservative and leads to loss of throughput, especially when traffic changes abruptly[6].

Unlike ARED, Re-ARED[9] aims to bring average queue size (avg) within its *target range* more *quickly* by adapting max_p aggressively. In this paper, we show that adaptation of max_p in Re-ARED algorithm may lead to instability in the average queue size, which if not eliminated, can significantly degrade the performance of the network. Hence, we propose Fast Adapting Random Early Detection (FARED) algorithm, a modification of Re-ARED, which efficiently adapts max_p to improve the overall performance of the network.

The remainder of the paper is organized as follows: Section 2 describes the effectiveness and limitations of Re-ARED. Section 3 provides details of design of FARED algorithm. Section 4 demonstrates the results and Section 5 concludes the paper.

2 Re-ARED: Effectiveness and Limitations

ARED's fixed and conservative approach of adapting max_p leads to degradation of throughput when level of congestion changes abruptly, especially in light and moderate traffic load scenarios. Re-ARED addresses the drawback of ARED and adapts max_p based on the ratio of the change in the average queue size that infers changes in the traffic load. Re-ARED algorithm as proposed in [9] is shown in Algorithm 1.

Algorithm 1. Re-ARED Algorithm

every interval seconds :
if $avg < target$ **and** $max_p \geq 0.01$ **then**
  decrease max_p
  $\beta = 1 - (0.17 \times \frac{target-avg}{target-min_{th}})$
  $max_p = max_p \times \beta$
end
else if $avg > target$ **and** $max_p \leq 0.5$ **then**
  increase max_p
  $\alpha = 0.25 \times max_p \times \frac{avg-target}{target}$
  $max_p = max_p + \alpha$
end