POSTER ABSTRACT

RED with dynamic thresholds for improved fairness

Vladimir Vukadinović and Ljiljana Trajković
School of Engineering Science
Simon Fraser University
Vancouver, BC, Canada
+1 604 291 3998
{vladimir, ljilja}@cs.sfu.ca

ABSTRACT

We investigate the fair bandwidth sharing among responsive and unresponsive traffic flows. When these flows compete for the same output link in a router, unresponsive flows tend to occupy more than their fair share of the link capacity. We propose a new active queue management algorithm named Random Early Detection with Dynamic Thresholds (RED-DT) that dynamically adapts queue parameters to achieve a more fair distribution of the link capacity.

Keywords

Fairness, TCP, UDP, active queue management, RED.

1. INTRODUCTION

Transmission Control Protocol (TCP) traffic represents vast majority of today's Internet traffic. TCP's congestion avoidance algorithm adapts sending rate based on congestion conditions in the network. Protocols having this property are called responsive. User Datagram Protocol (UDP) is the most commonly used protocol for real-time services. UDP flows are unresponsive because UDP does not react to network congestion. When congestion occurs, buffers could overflow and incoming packets will be discarded. TCP sender recognizes lost packets as a sign of network congestion and reduces its sending rate. In contrast, UDP senders do not have any knowledge about congestion and their sending rates will not be adjusted. Therefore, absence of congestion avoidance mechanism in UDP leads to unfair distribution of the available bandwidth between responsive and unresponsive flows.

One approach to solving the unfairness problem is to employ Active Queue Management (AQM) algorithms. The first and the most widely implemented AQM algorithm is Random Early Detection (RED) [2]. RED is unable to restrict unresponsive flows because it discards all incoming packets with a same probability. Since responsive and unresponsive flows react differently to packet losses (unresponsive flows do not react) this leads to an unfair distribution of link bandwidth. Certain AQM algorithms are able to identify and restrict unresponsive flows by preferentially discarding packets from these flows. Examples of such algorithms are Flow Random Early Detection (FRED) [4], Longest Queue Drop (LQD) [6], and CHOKe [5]. In this paper, we describe a new AQM algorithm named Random Early Detection with Dynamic Thresholds (RED-DT).

SAC '04, March 14-17, 2004, Nicosia, Cyprus.

2. RED-DT ALGORITHM

Output rate of a flow during congestion period can be inferred from its buffer occupancy. Greedy and unresponsive flows that use more than their fair share of bandwidth also use more than their fair share of the buffer space. In order to identify unresponsive and greedy flows, RED-DT maintains per-flow state for active flows. Flow is considered to be active if it has at least one packet in the queue. For each active flow, there is an entry in a flow table that contains the instantaneous queue size q^i, average queue size q^i_{ave}, and maximum drop probability $p^i_{\max}$. As in RED, instantaneous aggregate queue size q and average aggregate queue size q_{ave} are also monitored. Similar to RED, RED-DT maintains minimum and maximum thresholds, where $\min_{th} = 3\max_{th}$ [1]. However, in RED-DT, these thresholds are dynamically changed upon each packet arrival. If the new packet belongs to flow i, $\max_{th}$ is recalculated as:

$$\max_{th} = (1 - \alpha\, p^i_{\max})(B - q), \tag{1}$$

where B is the buffer size, $p^i_{\max}$ is maximum drop probability for flow i, and α is a constant discussed in the Section 3. RED-DT calculates drop probability for arriving packets as:

$$p = \begin{cases} 0 & if\ q^i_{ave} \le \min_{th} \\ \dfrac{q^i_{ave} - \min_{th}}{\max_{th} - \min_{th}}\, p^i_{\max} & if\ \min_{th} < q^i_{ave} < \max_{th} \\ 1 & if\ q^i_{ave} \ge \max_{th} \end{cases}. \tag{2}$$

Unlike RED, which compares the average aggregate queue size q_{ave} with thresholds, RED-DT compares average queue size of a particular flow q^i_{ave} with thresholds. When a packet is admitted to the queue, maximum drop probabilities $p^i_{\max}$ are updated as:

$$p^i_{\max} = \begin{cases} \max(1, p^i_{\max} + \delta) & if\ q^i_{ave} > \dfrac{q_{ave}}{N} \\ p^i_{\max} & if\ q^i_{ave} = \dfrac{q_{ave}}{N} \\ \min(p_{\min}, p^i_{\max} - \delta) & if\ q^i_{ave} < \dfrac{q_{ave}}{N} \end{cases}, \tag{3}$$

where δ is a constant increment, p_{min} is RED-defined parameter, and N is the number of active flows. In order to provide early

feedback to responsive flows, maximum drop probability $p^i_{\max}$ cannot be smaller than p_{min}. One choice for increment δ is $\delta=p_{min}$. Maximum drop probability $p^i_{\max}$ gradually increases for flows whose average queue size is larger than *1/N* of the average aggregate queue size q_{ave}. These flows are identified as potentially unresponsive and greedy. At the same time, their thresholds are decreased according to (1). When the average buffer occupancy of these flows decreases below q_{ave}/N, their maximum drop probability $p^i_{\max}$ gradually decreases and their thresholds increases. This mechanism aims to distribute the buffer space equally among active flows.

Recalculation of thresholds is the major complexity issue in RED-DT. Memory requirements for storing the per-flow states are limited to the size of the active flow table.

3. PERFORMANCE EVALUATION

We used ns-2 [7] to simulate RED-DT and compare its performance to RED [2], FRED [4], LQD [6], and CHOKe [5] that have been already implemented in ns-2. All simulation scenarios employ a common network topology shown in Fig. 1. To transfer FTP data, TCP New Reno sources have fixed packet size of 500 bytes. UDP sources use 500-byte packets to transfer constant bit-rate (CBR) streams. Default buffer size is 200 packets (80 ms on a 10 Mbps link), while UDP rate is 2.5 Mbps.

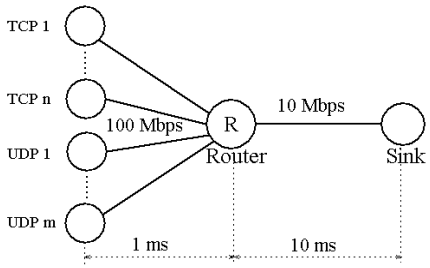

Figure 1. Simulated network topology.

In the first simulation scenario, 16 TCP flows compete with 4 UDP flows. Aggregate sending rate of the UDP flows varies from 5% to 120% of the link capacity. We observe average throughput achieved by each flow and calculate Jain's fairness index [3]. The values of the fairness index closer to 1 indicate a higher degree of fairness. Results are shown in Fig. 2 (left). They indicate that the fairness index of RED-DT does not depend on UDP rates. FRED performs well, albeit worse than RED-DT, while RED and CHOKe are unable to cope with the increase of the UDP load.

In the second simulation scenario, we evaluate the influence of the buffer size on RED-DT performance. As shown in Fig. 2 (middle), RED-DT performs well when buffer size varies from 40 ms to 200 ms. Hence, RED-DT is adaptable to variations in buffer sizes. FRED requires at least 120 ms of buffer space to achieve the same performance as RED-DT. LQD and RED do not improve their fairness even for large buffer sizes.

Parameter α in (2) is the most important parameter in the RED-DT algorithm. If $\alpha=0$, thresholds max_{th} and min_{th} do not depend on p_{max}. Larger α implies that flow *i*, identified as greedy, will be further restricted by decreasing its thresholds proportionally to $p^i_{\max}$. Fairness index for various α is shown in Fig. 2 (right). Good choices of α are between 0.3 and 0.6.

4. CONCLUSIONS

In this paper, we proposed a new active queue management algorithm named RED-DT. Its goal is to provide a fair bandwidth distribution for competing flows in a congested router. Simulation results and comparisons with the existing AQMs, such as RED, FRED, LQD, and CHOKe show better performances of RED-DT. RED-DT offers a trade-off between performance and complexity.

5. REFERENCES

[1] S. Floyd, "RED: discussions of setting parameters," Nov. 1997: http://www.icir.org/floyd/REDparameters.txt.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," in *IEEE Transactions on Networking*, vol. 1, no. 4, pp. 397-413, Aug. 1993.

[3] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling*. New York: John Wiley & Sons, 1991.

[4] D. Lin and R. Morris, "Dynamics of random early detection," in *Proc. of ACM SIGCOMM '97*, Cannes, France, Oct. 1997, pp. 127-137.

[5] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe: a stateless active queue management scheme for approximating fair bandwidth allocation," in *Proc. of IEEE INFOCOM'00*, Tel-Aviv, Israel, Apr. 2000, pp. 942-951.

[6] B. Suter, T. V. Lakshman, D. Stiliadis, and A. Choudhury, "Design considerations for supporting TCP with per-flow queuing," in *Proc. of IEEE INFOCOM'98*, San Francisco, CA, Apr. 1998, pp. 299-306.

[7] The Network Simulator - ns-2: http://www.isi.edu/nsnam/ns/.

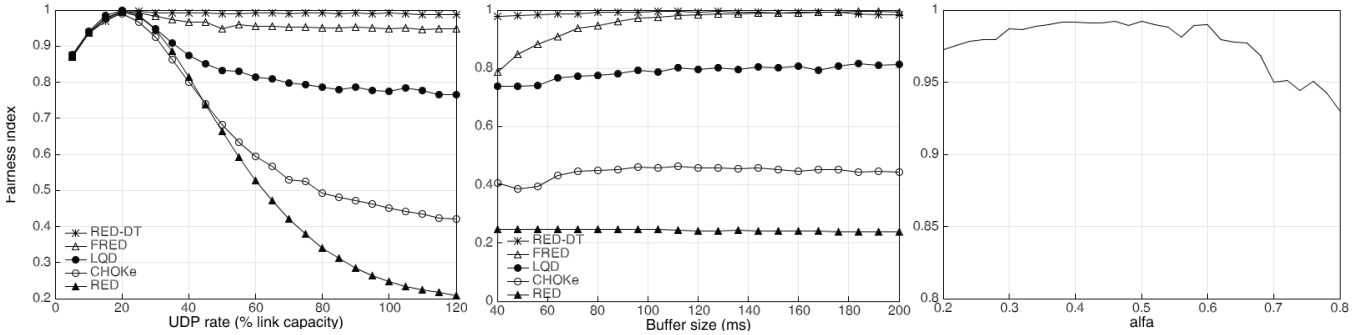

Figure 2. Fairness index for various UDP loads (left), buffer sizes (middle), and parameters α (right).