Journal of Network and Computer Applications 36 (2013) 858–869

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

On the use of a full information feedback to stabilize RED

Shahram Jamali [a,*], Seyyed Nasser Seyyed Hashemi [b], Amir Masoud Eftekhari Moghadam [c]

[a] Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran
[b] Young Researchers Club, Ardabil branch, Islamic Azad University, Ardabil, Iran
[c] Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

A R T I C L E I N F O

Article history:
Received 31 May 2012
Received in revised form
13 September 2012
Accepted 3 November 2012
Available online 29 November 2012

Keywords:
Active queue management
RED
Stability

A B S T R A C T

RED and most of its family algorithms use only the average queue length as a congestion meter. Since the average queue length considers only long-term behavior of the queue, these algorithms fail to see instantaneous changes of the queue length and hence their reaction to the congestion is not fast enough. In other words the feedback generated by using only the average queue length does not reflect the network congestion precisely and hence leads to a poor performance and stability. This paper solves this problem by designing a RED-based active queue management (AQM) algorithm, called FUF-RED that provides a Full Information Feedback. This algorithm not only considers the average queue length but also it takes into account growth rate of the instantaneous queue length to calculate its congestion feedback. The proposed algorithm is supported by a theoretical stability analysis which gives those feedback gains that guarantees the network stability. Extensive packet level simulations, done by using ns-2 simulator, show that the proposed algorithm outperforms existing AQM algorithms in terms of stability, average queue length, number of dropped packets and bottleneck utilization.

1. Introduction

With the rapid growth of the Internet, the increasing trends to new Internet application services and the proliferation of Internet users, the Internet's burden may go beyond the capacity of the Internet's communication and lead to Internet congestion. The network congestion control has been known as one of the critical issues for the last two decades and attracted a lot of attentions in these years (Jacobson, 1988; Analoui and Jamali, 2007; Jamali and Zahedi, 2010; Jamali and Analoui, 2011; Jamali and Eftekhari, 2011; Wang, 2005; Guo et al., 2008; Zheng and Wang, 2010; Hollot et al., 2002). The congestion control mechanisms in the Internet consist of the congestion window algorithms of transmission control protocol (TCP), running at end-systems, and active queue management algorithms running at the routers, seeking to obtain high network utilization, small amounts of queuing delay, and some degree of fairness among users (Analoui and Jamali, 2007; Jamali and Zahedi, 2010). RED (Floyd and Jacobson, 1993) is the most famous AQM control schemes explicitly introduced for congestion control that solves some major drawbacks of former approaches such as global synchronization. Due to its popularity, RED (or its variants) has been implemented by many router vendors in their products (e.g. Cisco used WRED, Cisco Systems). The basic idea of the RED is to detect the appearance of congestion by inspecting the average queue length at the routers. Before the real congestion take place, RED drops data packets with increasing probability when the average queue length lies between two thresholds (min_{th}, max_{th}) in order to inform sources about the coming congestion. As a consequence, when a TCP source finds out such preventive drops, it reduces the sending rate according to the additive increase multiplicative decrease (*AIMD*) algorithm inherent to the TCP protocol. In the recent years, various works have been done to enhance RED's performance (Floyd et al., 2001; Lim et al., 2002; Sun et al., 2003; Chen and Yang, 2009; Abbasov and Korukoglu, 2009). Some of these works are trying to increase RED's performance by improving its drop probability calculation function (Xiong et al., 2008, 2010; Liu et al., 2005; Zhou et al., 2006; Cho et al., 2008) and some other works improve it by dynamic tuning of its parameters such as minth and maxth (Jamali and Zahedi, 2010; Chen and Yang, 2009). Yousefi'zadeh et al. (2012) present the results of systematic study on optimal fine tuning of RED parameters and as result, they have identified the RED parameters locally minimizing the loss characteristic of the queue while satisfying an acceptable delay profile. A statistical approach is given in Barrera et al. (2011) to develop improved AQM mechanism and decrease queue fluctuations. It denotes that the statistical characteristics of packet markings provide a performance bound of AQM in relation to the queue variance. So it proposes a simple marking strategy to

* Corresponding author. Tel.: +98 4515 513917; fax: +98 4515510141.
E-mail addresses: jamali@iust.ac.ir (S. Jamali), n.s.hashemi@qiau.ac.ir (S.N. Seyyed Hashemi), eftekhari@qiau.ac.ir (A.M. Eftekhari Moghadam).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.11.002

reduce the queue's variance by one order of magnitude from that attained with probabilistic drops. An improved adaptive-RED AQM was proposed in Zhang et al. (2011) which imposes nonlinear smooth for dropping function of RED by using the membership function of the ascend demi-cauchy of fuzzy distribution. In the proposed method, the maximum drop probability will be dynamically adjusted by average queue length. Another RED-based approach is proposed in Chen et al. (2012) to design a new AQM which integrates RED and fuzzy proportional integral derivative (FuzzyPID) approach. This algorithm controls the buffer queue around a target value. Existing AQMs are categorized in Suzer et al. (2012) as ad-hoc event-driven control and time-driven feedback control approaches based on control theory. A model predictive (MP) control as AQM algorithm was proposed in Wang et al. (2012). The proposed algorithm predicts future queue length in the data buffer and according this prediction, early packets at the router are dropped to keep the queue length around desired value. The algorithm proposed in Tahiliani et al. (2012) dynamically varies maximum drop probability based on the level of traffic load. This algorithm uses only the average queue length as its congestion indicator.

One of the major problems for the RED is instability of its queue length. Instability can cause three problems. First, it increases jitters in source. Second, it subjects short-duration connections, which are typically delay and loss sensitive, to unnecessary delay and loss. Finally, it can lead to under-utilization of network links if queues jump between empty and full (Wang, 2005). We believe that instability of RED queue length has roots in this fact that RED and most of its family algorithms use only average queue length as the congestion meter. Since, this approach ignores instantaneous changes of the queue length, it results in many oscillations in the queue length. This paper aims to solve this problem by designing a RED-based active queue management algorithm in which the network congestion is measured in terms of both the instantaneous and the average queue length. For this purpose it presents a novel packet drop probability function that not only considers the average queue length in its calculations but also it takes into account growth rate of the instantaneous queue length.

The rest of the paper is organized as follows. Section 2 briefly presents a review over some existing RED-based queue management algorithms. Section 3 presents idea and principals behind the FIF-RED algorithm. Stability of FIF-RED algorithm will be discussed in Section 4. Section 5 evaluates performance of FIF-RED comparing to some other queue management algorithms through a simulative study in ns-2 environment (NS-2) and finally Section 6 presents concluding remarks.

2. Preliminaries: drop probability function in RED family algorithms

In this section we take a look at drop probability function of various RED family algorithms. These algorithms calculate their drop probability based on RED's basic four parameters, namely, minimum queue length threshold (min_{th}), maximum queue length threshold (max_{th}), maximum packet dropping probability (max_P) and w_q, which is the weighting factor for computing the average queue length. The drop probability of RED can be illustrated by using the following equation:

$$P(t)=\begin{cases}0 & avg(t)\le min_{th},\\ \frac{avg(t)-min_{th}}{max_{th}-min_{th}}max_p & min_{th}\le avg(t)\le max_{th},\\ 1 & avg(t)\ge max_{th}\end{cases} \tag{1}$$

In the equation above, $P(t)$ is drop probability at time t and $avg(t)$ is the average queue length which it is estimated by RED at time t. According to this equation, RED performance is very sensitive to the traffic intensity and the mentioned parameters. Various algorithms that have been designed based on RED algorithms typically differ only in how they tune their parameters.

Now we compare different RED-based algorithms from their dropping function point of view. Figure 1 describes schematically how different algorithms calculate their dropping probabilities based on their queue length. As shown in Fig. 1a, when average queue length of RED exceeds max_{th}, drop probability jump to 1 in order to avoid a persistently full queue. So, when the average queue becomes larger than max_{th}, RED often does not perform well and resulting in significantly decreased throughput and increased dropping rates. This weakness has been improved in Gentle-RED (Lim et al., 2002). In Gentle-RED the packet dropping probability increases linearly from max_p to 1 when average queue length increases from max_{th} to $2*max_{th}$ (see Fig. 1b). With the nonlinear packet dropping function in Nonlinear-RED (Zhou et al., 2006) packet dropping becomes gentler than former schemes at light traffic but more aggressive at heavy load. Nonlinear-RED uses the nonlinear quadratic function shown in Eq. (2) to drop packets, where max'_p represents the maximum packet dropping probability of Nonlinear-RED (see Fig. 1c):

$$P(t)=\begin{cases}0 & avg(t)\le min_{th},\\ \left(\frac{avg(t)-min_{th}}{max_{th}-min_{th}}\right)^2 max'_p & min_{th}\le avg(t)\le max_{th},\\ 1 & avg(t)\ge max_{th}\end{cases} \tag{2}$$

In Chen and Yang (2009) an auto-parameterization RED (*AP-RED*) is designed to provide a systematic algorithm for tuning four key RED parameters as function of network traffic conditions (see Fig. 1d). On the other hand, some weaknesses of RED that affects its performance have been corrected in Effective-RED (Abbasov and Korukoglu, 2009). The main idea of Adaptive-RED (Floyd et al., 2001) can be easily summarized as setting w_q automatically based on the link speed and adapting maxp in response to measured queue length. Similarly, Tan et al. (2006 and Zhang et al. (2009) have provided guidelines for adjusting only one of the RED parameters in response to the changing network conditions.

In fact, all the above mentioned schemes have been presented to improve the responsiveness of RED dynamic and stabilize the queue length around the target value. As we saw the original RED and its variants use average queue length as a congestion indicator to trigger packet dropping. This imposes function's outcome heavy in front of traffic load changes. Few works such as Xiong et al. (2010) have considered the instantaneous queue length in drop probability function instead of the average queue length. Although the instantaneous queue length exhibits rapid response to burst congestion, it may cause instability in some conditions. In the next section we propose a dropping function in which both average queue length and instantaneous queue length are considered jointly to design a more stable and high performance active queue management scheme.

3. The FIF-RED algorithm

As discussed above, all the RED family algorithms use either the average queue length or the instantaneous queue length as the congestion indicator. Using a weighted averaging method avoids fast reaction to burst traffic and regards only long-term trends. In contrast, instantaneous queue length has rapid response to burst congestion, but its side effect is its possible instability. We believe that a combination of average and instantaneous queue lengths can make a better feedback that returns