Computer Communications 32 (2009) 1525–1530

Contents lists available at ScienceDirect

Computer Communications

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

The mechanism of adapting RED parameters to TCP traffic

Wu Chen, Shuang-Hua Yang *

Department of Computer Science, Loughborough University, Loughborough LE11 3TU, UK

ARTICLE INFO

Article history:
Received 8 August 2008
Received in revised form 13 May 2009
Accepted 13 May 2009
Available online 27 May 2009

Keywords:
RED
Congestion control
Parameter tuning
Control theory

ABSTRACT

Random Early Detection (RED) can stabilize the instantaneous queue size at a router by controlling the average queue size within a given target. In this case, RED can achieve high throughput in the routers. However, the average queue size is quite sensitive to network scenarios and it is difficult to adapt RED parameters to changing network traffic. In this paper we use a previously developed dynamic model of TCP behavior together with a linear feedback model of TCP/RED to analyze and design a mechanism for RED parameter tuning in response to changing network conditions like traffic load, link capacity and round-trip time. Even though the values of four key RED parameters are determined by varying network conditions, they can be tuned independently without consideration of the interactions among these RED parameters. Simulation results show that this mechanism can keep the instantaneous queue size stable and maintain high link utilization in a wide variety of network conditions.

1. Introduction

Internet congestion control plays an important role in the Internet to ensure good network performance. It comprises two main parts: TCP and Active Queue Management (AQM). Among many AQM algorithms proposed over the last decade, RED has been widely implemented in major commercial routers, especially in edge routers [1,2]. It is seldom deployed in core routers because they typically do not run congested links [3]. One of the main goals of RED is to stabilize the instantaneous queue size by controlling the average queue size within a given target, while maintaining high link utilization [4]. However, the average queue size varies with traffic load as well as round-trip time and link capacity, and parameterizing RED to obtain good performance under variable congestion scenarios is very difficult. Such difficulties discourage network administrators from activating RED in their routers [5]. Certainly, there are many parameter tuning techniques for RED proposed in the literature [6–12], but they were either developed on the basis of empirical investigations and analysis, or are only applicable under certain assumptions. In particular, the authors in [6–9] provided guidelines for adjusting only one of the RED parameters in response to the changing network conditions. So these guidelines are applicable only under a narrow range of round-trip times and link capacities. As demonstrated in other AQM or AQM-based approaches, such as the Proportional Integrator (PI) controller [11], Random Exponential Marking (REM) [13] and BLUE [14], parameter setting still remains a critical unsolved problem. In addition, the Web performance of these AQMs is not as good as Adaptive RED (ARED) in [6] if dropped packets are used as indications of congestion [15], and their transient response would become much slower when network scenarios change dynamically [16].

In [12] the authors proposed a method to set initial RED parameters from a Control Theoretic perspective. However, they did not illustrate how to adapt these RED parameters to changing network scenarios. By improving their Control Theoretic analysis of TCP/RED systems, we developed an Auto-Parameterization RED (AP-RED) to provide a simple, scalable and systematic algorithm for tuning four key RED parameters as a function of network traffic conditions of link capacity, round-trip time, and the number of TCP flows. Theoretic analysis and nonlinear simulations using a *ns-2* simulator [17] have demonstrated that it is robust, adaptive to TCP dynamics, and produces desirable transient performance.

The rest of the paper is organized as follows. Section 2 illustrates a pre-developed nonlinear dynamic model of TCP and a linear feedback model of TCP/RED system. Section 3 describes AP-RED and presents stability analysis. Section 4 introduces the calculation method for network parameters. Simulation results to validate the algorithm are presented in Section 5. Finally, we conclude in Section 6.

2. Model

In [18], a nonlinear dynamic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis. By ignoring timeout mechanism, Hollot et al. [12] uses a simplified version to describe the model by the following differential equations:

---

* Corresponding author. Tel.: +44 1509 635670; fax: +44 1509 635722.
E-mail addresses: w.chen2@lboro.ac.uk (W. Chen), s.h.yang@lboro.ac.uk (S.-H. Yang).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.05.008

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t))$$
$$\dot{q}(t) = \frac{W(t)}{R(t)}N(t) - C \tag{1}$$

where $\dot{x}$ denotes the time-derivative of x and

$W \doteq$ expected TCP window size(packets);
$q \doteq$ expected queue size(packets);
$R \doteq$ round-trip time(seconds);
$C \doteq$ link capacity(packets/second);
$N \doteq$ number of TCP flows;
$p \doteq$ probability of packet mark/drop.

According to these equations, Hollot et al. [12] further describes the behavior of $\delta W \doteq W - W_0, \delta q \doteq q - q_0$, and $\delta p \doteq p - p_0$ in a linear feedback control model of TCP/RED by linearizing variables (W, q, p) at its equilibrium point (W_0, q_0, p_0). The linear control model is depicted in Fig. 1.

In the above model $P_{tcp}(s)$ denotes the linearized TCP dynamics, $P_{queue}(s)$ denotes the queue dynamics, e^{-sR} denotes the delay term and $C_{red}(s)$ denotes RED control strategy. They were given by the following equations:

$$\begin{cases} P_{tcp}(s) = \frac{\frac{RC^2}{2N^2}}{s+\frac{2N}{R^2C}} \\ P_{queue}(s) = \frac{\frac{N}{R}}{s+\frac{1}{R}} \\ C_{red}(s) = \frac{L_{red}}{s/K+1} \end{cases} \tag{2}$$

where

$$L_{red} = \frac{p_{\max}}{\max_{th} - \min_{th}};$$
$$K = -\frac{\log_e(1-\alpha)}{T_s};$$

$p_{\max} \doteq$ maximum drop probability;
$\max_{th} \doteq$ maximum threshold;
$\min_{th} \doteq$ minimum threshold;
$\alpha \doteq$ average queue weight;
$T_s \doteq$ sampling interval.

3. AP-RED algorithm and stability analysis

3.1. The algorithm

The objective of AP-RED is to stabilize the instantaneous queue size by maintaining the average queue size within the limits of $\max_{th}$ and $\min_{th}$ in widely varying network scenarios. The rationale behind setting $\max_{th}$ and $\min_{th}$ as a fraction of bandwidth-delay product is to follow the theoretic analysis and rule-of-thumb in accommodating bursty traffic [6,7,19]. The rational behind setting $p_{\max}$ is to achieve a desirable equilibrium point based on theoretic analysis. The rationale behind setting α is to maintain the stability of the system from a Control Theoretic perspective.

Thus, consider the initial RED parameters $\alpha_0, p_{\max 0}, \max_{th0}$, and $\min_{th0}$ that stabilize the queue size under the initial network scenario of TCP loadN_0, round-trip time R_0 and link capacity C_0. When the network scenario varies we present the following algorithm for adjusting RED parameters.

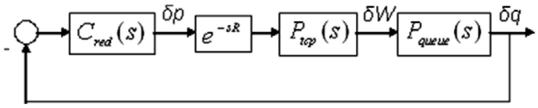

Fig. 1. Feedback control model of TCP/RED system.

$$\max_{th} = k_r k_c \max_{th0} \tag{3}$$
$$\min_{th} = k_r k_c \min_{th0} \tag{4}$$
$$p_{\max} = \left(\frac{k_n}{k_r k_c}\right)^2 p_{\max 0} \tag{5}$$
$$\alpha = \begin{cases} \frac{k_n}{(k_r k_c)^2}\alpha_0 & N \leqslant RC/2 \\ \frac{1}{k_r k_c}\alpha_0 & N > RC/2 \end{cases} \tag{6}$$

where

$k_r = R/R_0;$
$k_c = C/C_0;$
$k_n = N/N_0$

and constraint $p_{\max}$ within the range [0.01,0.5]

i.e. if $p_{\max} > 0.5 \quad p_{\max} = 0.5$
if $p_{\max} < 0.01 \quad p_{\max} = 0.01$

Compared with the automatic setting of average queue weight α based on link capacity in [6], our tuning algorithm of α is based not only on link capacity, but also on round-trip time and traffic load. As α is adapted to congestion scenarios, a larger α can improve transient response while a smaller α provides a sufficient stability margin [12]. When $\max_{th}$ and $\min_{th}$ are reduced according to the changing network parameters, the queuing delay is decreased correspondingly. In contrast, the average queue size can still be stabilized within a given target range by using a larger $\max_{th}$ and $\min_{th}$ when round-trip time increases or link capacity increases.

3.2. Determining the equilibrium point

The purpose of adjusting $p_{\max}$ is to keep the queue size at a desirable equilibrium point. At a new equilibrium point where $\dot{W} = 0$ and $\dot{q} = 0$, from (1) we have

$$W^2 p = 2 \quad \text{and} \quad W = \frac{RC}{N}$$

Then we obtain

$$p = \frac{q - \min_{th}}{\max_{th} - \min_{th}} p_{\max} = 2\left(\frac{N}{RC}\right)^2 \tag{7}$$

Hence at the equilibrium point the drop probability is determined by network scenario parameters rather than other RED parameters.

Similarly, at the initial equilibrium point q_0 we obtain

$$p_0 = \frac{q_0 - \min_{th0}}{\max_{th0} - \min_{th0}} p_{\max 0} = 2\left(\frac{N_0}{R_0 C_0}\right)^2 \tag{8}$$

Since we tune $p_{\max}$ in terms of (5), from (8) and (9) we have

$$q = k_r k_c q_0$$

Thus the equilibrium point q moves in proportion to the $\max_{th}$ and $\min_{th}$. It still stays between $\max_{th}$ and $\min_{th}$ when network condition changes.

3.3. Stability analysis

3.3.1. Stability proposition

In this section we derive a simplified version of stability proposition based on the analysis given in [12].