ELSEVIER

Computer Communications 25 (2002) 1736–1750

computer communications

www.elsevier.com/locate/comcom

A linear system analysis of RED

James Aweya*, Michel Ouellette, Delfin Y. Montuno

Nortel Networks, Systems Architect, P.O. Box 3511, Station C, Ottawa, Ont., Canada K1Y 4H7

Received 5 September 2001; revised 17 February 2002; accepted 27 February 2002

Abstract

The random early detection (RED) algorithm proposed by Floyd and Jacobson [IEEE/ACM Trans. Networking 1 (4) (1993) 397] is a simple and quite effective mechanism for active queue management and as a result has been widely implemented in Internet routers. While RED certainly performs better than drop-tail, its parameters are difficult to tune to give good performance under different congestion scenarios. Also, studies have shown that RED does not stabilize the queue lengths in routers, because its equilibrium queue length strongly depends on the number of active TCP connections. As a result numerous variants of RED have been proposed to work around the performance problems of RED. In this paper, using inferences drawn from a linear systems analysis, we provide an insight into why RED is difficult to tune. The same linear systems analysis is used to show that the DRED algorithm proposed in [Computer 36 (2–3) (2001) 203; Comput. Commun. 24 (12) 2001 1170] overcomes the performance limitations of RED [IEEE/ACM Trans. Network. 1 (4) (1993) 397]. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: TCP; Congestion control; Active queue management; Random early detection; Control theory

1. Introduction

The idea behind active queue management (AQM) is to detect incipient congestion in routers and switches early and convey congestion notification to the end-hosts, allowing them to back off before queue overflow and sustained packet loss occur. One form of AQM recommended by the IETF for deployment in the Internet is RED [1,4]. Studies have shown that RED does not stabilize queue lengths in routers, because its equilibrium queue length strongly depends on the number of active TCP connections [5]. The dynamics of TCP changes when the number of active connections changes, and RED exhibits sensitivity to the specific traffic scenario in the network. The performance of RED can be tuned by adjusting its parameters but such a process requires knowledge of network traffic load. Configuring RED's parameters to optimize network performance is an open issue and is still under active research.

Using inferences from simple linear system analysis, we explain here why RED is difficult to tune and why RED is not able to stabilize router queues over a wide range of congestion scenarios. The interest in understanding the tuning issues in RED has become very important and stems mainly from the following factors:

- Currently, RED is the only algorithm widely implemented in routers, switches, etc.
- System and network engineers want to know how to configure/tune RED.
- System and network engineers want to know if they should even deploy RED (e.g. see Ref. [6]).
- Is there a need to look at the AQM problem from an entirely different perspective?

The discussion here is motivated by ideas that are familiar in linear system analysis and control theory. This allows us to make parallels and use insights developed in these fields. We do acknowledge that the models used in the discussions are a rough representation of the AQM process, however, they are adequate enough to help us make the necessary inferences about the behavior of the algorithms under discussion. The linear systems analysis is also used to show that the DRED algorithm proposed in Refs. [2,3] overcomes the performance limitations of RED [1]. Including a discussion on DRED here is based on the desire to answer the following question: could the linear system approach on which the development of DRED is based also offer an insight into the tuning difficulties of RED?

DRED uses a simple feedback control approach to randomly discard packets with a load-dependent probability when a buffer in a router becomes congested. Over a wide range of load levels, DRED is able to stabilize a router

* Corresponding author. Tel.: +1-613-763-6491; fax: +1-613-765-0678.
E-mail address: aweyaj@nortelnetworks.com (J. Aweya).

0140-3664/02/$ - see front matter
PII: S0140-3664(02)00105-6

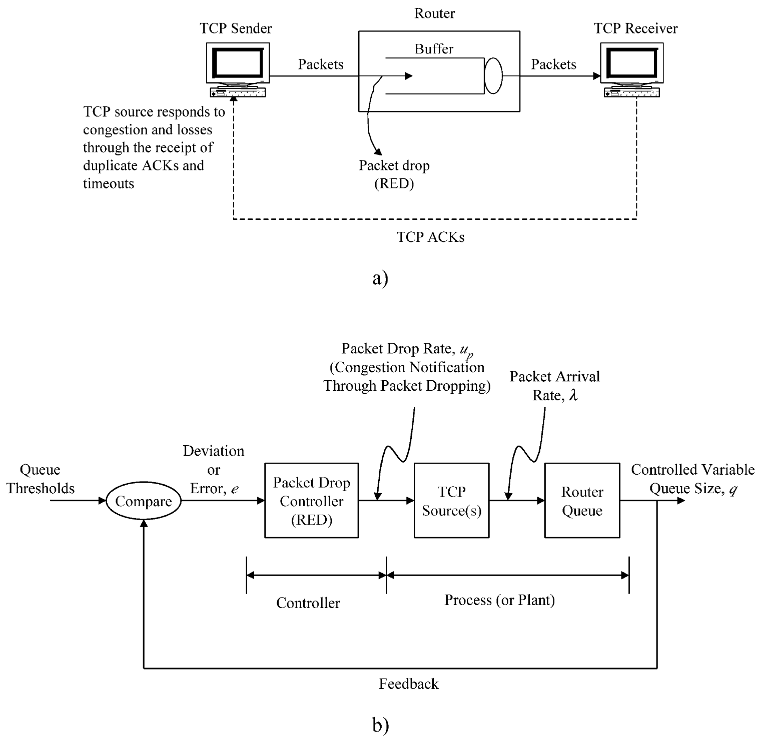

Fig. 1. (a) TCP congestion avoidance through random early detection, (b) TCP congestion avoidance as a closed-loop feedback control system.

queue occupancy at a level independent of the number of active TCP connections. *DRED maintains the average queue size close to a predetermined threshold, but allows transient traffic bursts to be queued without unnecessary packet drops*. This is done without estimating the number of active TCP connections or flows and also without collecting or analyzing state information on individual flows.

A detailed discussion and investigation of RED and DRED including simulations with different network configurations and sources with different round-trip times (RTTs), traffic load, link rates, and sensitivity studies are given in Refs. [2,3]. A detail comparison of DRED with RED [1] and some recently proposed AQM algorithms (BLUE [7], SRED [8]) for short-lived and long-lived flows is given [9]. Thus, the simulation studies included here serve only to explain/verify the observations made from the linear analysis. We also note here that the previously published papers [2,3,9] focus on the design of DRED as a new alternative to RED [1] and do not present analytic discussions (based on linear system theory) on the tuning difficulties of RED. These analytic discussions are the focus of this paper.

2. The RED algorithm

2.1. Basic description

A RED module controls congestion by randomly dropping packets with a probability u_p, which constitute a signal to applications (TCP sources) to reduce their sending rate as illustrated in Fig. 1(a). Typically, a RED module consists of a control function that computes a drop probability (or rate) as a function of some monitored variables in the router queues (e.g. average queue size). A generalized feedback control model that embodies the above elements is shown in Fig. 1(b).

RED [1] uses an exponentially weighted moving

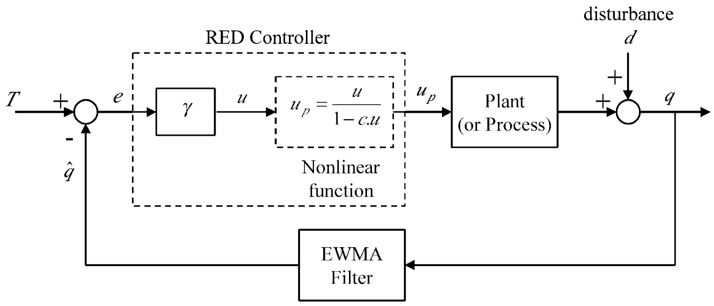

Fig. 2. A discrete-time representation of RED as pseudo-proportional controller.