Computer Networks 88 (2015) 1–11

Contents lists available at ScienceDirect

Computer Networks

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

Capacity and token rate estimation for networks with token bucket shapers

Ertong Zhang, Lisong Xu*

Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, United States

ARTICLE INFO

Article history:
Received 3 December 2014
Revised 24 April 2015
Accepted 24 May 2015
Available online 9 June 2015

Keywords:
Token bucket
Capacity estimation
Computer and network virtualization

ABSTRACT

Many cloud computing applications are bandwidth-intensive, and thus the cloud bandwidth information is important for their tenants to manage and troubleshoot the application performance. However, current bandwidth estimation methods face great challenges with clouds. One important reason is that traffic shapers such as token bucket shapers are widely used as a building block for rate limiting in clouds. In this paper, we propose two methods called *NarrowLinkCapacity* and *NarrowTokenRate* to actively estimate the capacity and token rate, respectively, of a path in a network with potentially multiple token bucket shapers. The capacity of a path is the slowest link capacity among all links on the path, and the token rate of a path is the slowest token rate among all token bucket shapers in the path. They are two important path properties determining the average rate of a train of packets on a path. Our evaluation results show that our methods work well under a variety of network conditions.

1. Introduction

Cloud computing is transforming a large part of IT industry, as evidenced by the increasing popularity of public cloud computing services, such as Amazon Web Service [1], Google Cloud Platform [2], Microsoft Windows Azure [3], and Rackspace Public Cloud [4]. Many cloud computing applications are bandwidth-intensive, such as MapReduce applications and high performance computing applications, and thus the network bandwidth information of clouds is important for their tenants to manage and troubleshoot the application performance. For example, if the bandwidth information can be estimated, the application performance can be improved by appropriately placing application tasks on virtual machines [5].

Bandwidth estimation methods, such as *pathrate* [6], *capprobe* [7], *tailgater* [8], *pathload* [9], and *spruce* [10], have been successfully used to estimate the capacity and available bandwidth information of a network path in the traditional Internet. However, they face great challenges with clouds. One important reason is that traffic shapers are widely used as a basic building block for rate limiting [11] in clouds, however, traffic shapers interfere with the probing packets of bandwidth estimation methods.

In this paper, we study token bucket shapers that are a basic type of traffic shapers. Token bucket shapers have been widely used in virtualization software such as VMWare [12] and Xen [13], cloud computing platforms such as Amazon EC2 [1], large-scale virtual networks such as PlanetLab [14], software switches such as Open vSwitch [15] and OpenFlow Software Switch [16], and data center resource management schemes such as Seawall [17]. A popular type of token bucket shapers is the Token bucket filter (*tbf*) provided in Linux, which regulates traffic according to a token rate and a burst size. *tbf* is the building block of more advanced token bucket shapers, such as the hierarchy token bucket (*htb*) in Linux that regulates traffic using multiple *tbf* shapers and allows token borrowing among different shapers. In this paper, we focus on *tbf* and *tbf*-like shapers.

* Corresponding author. Tel.: +1 4026176135.
E-mail addresses: ezhang@cse.unl.edu (E. Zhang), xu@cse.unl.edu, xu@unl.edu (L. Xu).

http://dx.doi.org/10.1016/j.comnet.2015.05.013
1389-1286/© 2015 Elsevier B.V. All rights reserved.

The *contribution* of this paper is that we propose two methods to actively estimate the capacity and token rate, respectively, of a path in a network with potentially multiple *tbf* or *tbf*-like shapers. Specifically, we propose a method called *NarrowLinkCapacity* to estimate the capacity of a path, which is the slowest link capacity among all links in the path. It is an important property of the path, because it determines the average rate of a short train of packets. We also propose a method called *NarrowTokenRate* to estimate the token rate of a path, which is the slowest token rate among all token bucket shapers in the path. It is another important property of the path, because it determines the average rate of a long train of packets.

The rest of the paper is organized as follows. We first summarize related work in Section 2, and then introduce the token bucket shapers in Section 3. The design goals of our proposed methods are given in Section 4. *NarrowLinkCapacity* and *NarrowTokenRate* are described in Sections 5 and 6, respectively. The evaluation results are shown in Section 7, and finally the paper is concluded in Section 8.

2. Related work

Network bandwidth estimation methods can be classified into two categories: capacity estimation and available bandwidth estimation. Capacity estimation methods can be further classified into two classes: (1) methods to estimate the capacity of a path, such as *bprobe* [18] , *pathrate* [6], and *capprobe* [7], which mainly use the dispersion of two consecutive probing packets; and (2) methods to estimate the capacity of each individual link in a path, such as *pathchar* [19] and *tailgater* [8]. Available bandwidth estimation methods further fall into two classes: (1) probe gap model (PGM), such as *spruce* [10] and *IGI/PTR* [20], which use the initial and final time gap information of probing packets; and (2) probe rate model (PRM), such as *pathload* [9] and *pathchirp* [21], which are based on self-induced congestion.

There are very few related works on bandwidth estimation in networks with token bucket shapers. Khandelwal et al. [22] study the accuracy of available bandwidth estimation methods, such as *pathload*, on Amazon EC2, and they find that the current methods are "*un-suitable for bandwidth estimation in data center networks*". Lakshminarayanan et al. [23] measure the impact of token bucket shapers on bandwidth estimation methods in broadband access networks, and they conclude that "*both capacity and available bandwidth measurement are challenging because of the dichotomy between the raw link bandwidth and the token bucket rate*".

The closest work is *shaperprobe* developed by Kanuparthy and Dovrolis to measure the token bucket characteristics as a traffic shaping service in residential ISP networks [24]. *shaperprobe* detects a level shift in the measured rates, and estimates the token rate as the median rate after the level shift. Our work is different from *shaperprobe* in that we consider general networks whereas *shaperprobe* mainly considers residential ISP networks. There are two important differences between token bucket shapers in general networks and in residential ISP networks. First, token bucket shapers in residential ISP networks usually have bigger burst sizes (e.g., 5–10 MB), and as a result the path capacity can be estimated using existing capacity estimation methods. Second, there is no or very little background traffic competing with the probing packets at a token bucket shaper in residential ISP networks, and thus the token rate can be relatively easily estimated.

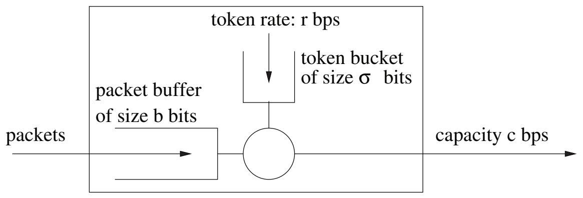

Fig. 1. A *tbf* or *tbf*-like shaper with four parameters (r, σ, c, b).

3. Networks with token bucket shapers

In this section, we introduce *tbf* and *tbf*-like shapers, and discuss their impact on the dispersions of a train of packets.

3.1. *tbf and tbf-like shapers*

Fig. 1 illustrates a *tbf* or *tbf*-like shaper. It consists of two buffers: a token bucket and a packet buffer. Tokens are generated and placed into the token bucket at a rate of r bits per second (bps). The token bucket can hold up to σ bits of tokens, and any newly generated token will be discarded if the token bucket is full. When a packet of size s bits arrives at the shaper, if there are at least s bits of tokens available, the packet is immediately transmitted to the outgoing link at its capacity c bps and at the same time s bits of tokens are consumed from the token bucket. Otherwise, the packet will be queued in the packet buffer until there are at least s bits of tokens available.

A *tbf* or *tbf*-like shaper is described by four parameters (r, σ, c, b): (1) token rate r bps, (2) burst size σ bits (also called token bucket size), (3) capacity c bps, and (4) packet buffer size b bits. Note that, $r \le c$. In addition, $\sigma \ge \sigma_{\min} = \text{MTU}$, and this ensures that a packet with the maximum transmission unit (MTU) size can pass though a token bucket. Since the typical MTU is 1500 bytes, we have $\sigma_{\min} = 1500 \times 8$ bits. For example, we observe that a PlanetLab [14] node sets its burst size σ to 1600×8 bits.

Considering a case where the token bucket is full, and a train of packets each of size s bits arriving at the shaper at rate $\lambda > r$. In this case, the first K packets will be transmitted at rate $\min(c, \lambda)$, where K is given in Eq. (1). Note that $K \ge \sigma/s$, because new tokens are being generated during the transmission of the first σ/s packets. After the first K packets, the token bucket becomes empty, and thus the remaining packets will be throttled by token rate r.

$$K = \begin{cases} \lfloor (\sigma/s - r/c)/(1 - r/c) \rfloor, & \text{if } \lambda \ge c \\ \lfloor (\sigma/s - r/\lambda)/(1 - r/\lambda) \rfloor, & \text{if } r < \lambda < c \end{cases} \quad (1)$$

3.2. *Impact on packet dispersions*

We use the following two simple networks to show the impact of token bucket shapers on a train of packets.

- *Network 1*: A one-hop network without any shaper. The link capacity between the sender and the receiver is c.