Computer Communications 53 (2014) 26–36

Contents lists available at ScienceDirect

Computer Communications

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

A new model-based clock-offset approximation over IP networks

Edmar Mota-García, Rogelio Hasimoto-Beltran *

Center for Research in Mathematics (CIMAT), Jalisco s/n, Col. Mineral de Valenciana, Guanajuato, Gto., Mexico

ARTICLE INFO

Article history:
Received 8 September 2013
Received in revised form 8 July 2014
Accepted 12 July 2014
Available online 30 July 2014

Keywords:
Gamma distribution
One-way delay
Clock-offset estimation
Network-time protocol

ABSTRACT

Having in mind multimedia systems applications, we propose a novel model-based approach to estimate the clock-offset between two nodes on the Internet. Different than current clock-offset schemes in the literature, which are iterative in nature, our scheme is aimed at getting a good non-iterative clock-offset estimation in real time (in the order of milliseconds). In our clock-offset estimation approach, the One-Way Delay (OWD) measurements are modeled with a shifted gamma distribution representing the current state of the probing link. By using the QQ-probability plot technique and linear regression model, we estimate the (shift parameter or) minimum value of the gamma distribution with probability zero. This estimated value represents the clock offset plus network propagation and transmission delay (queuing delay has already been eliminated) for the corresponding receiving path. End nodes exchange their corresponding minimum estimates and get an improved final clock offset estimate considering the network path asymmetries. Based on real experiments, we show that our scheme provides an extremely fast clock-offset estimation with lower RMSE and superior stability than NTP and current NTP-like state of the art methodologies in the literature Jeske and Sampath (2003), Choi and Yoo (2005), Adhikari et al. (2003), Tsuru et al. (2002). Moreover, our proposed scheme is non-intrusive (no kernel programming needed), easy to implement, and targeted as part of more complex real-time multimedia distribution protocols requiring a fast and reliable OWD estimates.

1. Introduction

One-Way Delay (OWD), end-to-end delay, or One-Way Transit Time (OTT) represents one of the most important Quality-of-Service (QoS) parameter to consider in current communication network applications such as multimedia streaming, distributed event time-ordering (distributed online-games, multiparty teleconference), routing, transport/application protocol design, network management and performance. The estimation of the OWD generally requires clock synchronization between end-nodes or hosts, which still is a fundamental problem in global or IP networks. The problem is even exacerbated for real-time multimedia applications (audio/video streaming, application-level multicast routing, distributed online-games, etc.), for which the OWD must be estimated promptly. A common approach for clock synchronization or clock offset estimation is the exchange of packets probes with send/receive time stamps in a synchronous send-and-wait for reply scheme, as performed by the NTP standard protocol [8] and current NTP improvements [19,1,7,18]. These schemes are iterative in nature (current estimate is an improved version of previous estimate) requiring long sampling periods (from minutes to hours) to provide an accurate clock offset estimate. The long probing periods makes them vulnerable to clock artifacts such as drift and skew. Even though clock artifacts can be easily eliminated [15], current NTP-like schemes still need to deal with long sampling periods to obtain the long-term minimum delay to estimate the OWD. Long probing may not be a problem for static nodes (network servers working on some specific task) such as routers over the Internet, web servers, Domain Name Servers (DNS). For dynamic nodes (end-systems or user nodes) requiring fast clock synchronization, long probing is not an acceptable option (users are not willing to wait for hours for their clocks to be synchronized before initiating a videoconference or distributed game).

Knowledge of the OWD is of great benefit during the activation of a multimedia communication session. In robust communications for example, it helps in several decision making process, such as: (a) the use of data re-transmission (ARQ) or data protection in the form of Forward Error Correction (FEC) or both in the case of transmission errors; (b) when ARQ is selected, the knowledge of the OWD further improves channel-bandwidth usage since only

* Corresponding author. Tel.: +52 473 732 7155; fax: +52 473 732 5749.
E-mail addresses: edmar@cimat.mx (E. Mota-García), hasimoto@cimat.mx (R. Hasimoto-Beltran).

http://dx.doi.org/10.1016/j.comcom.2014.07.006
0140-3664/

those packets that can make it on time at the receiver-end are retransmitted [21]; (c) selection of the receiving buffer length, (d) the application level of interactivity, (e) congestion control strategies (increase compression when detecting an increment in the OWD before information is lost). In the case of a multi-party distributed game, fast clock synchronization can be further used by receivers to assess the temporal order each stream was sent out from different senders. Similarly, fast offset estimation is also needed in applications such as Internet-TV and multi-party video-conferencing. They often require *Application-Level Multicast* (*ALM*) [22], in which IP-multicast related functionalities have been moved to user-nodes (this is due to difficulties in the commercial deployment of IP multicast). An ALM topology consists of unicast connection between user-nodes satisfying some constrains; among the most important is the overall transmission *latency or delay*. To get the optimal data distribution path visiting all user-nodes, the OWD between every pair of user–nodes must be known beforehand (not possible under current clock-synchronization schemes). Things get even worse when user-nodes are allowed to join and leave the multimedia session randomly; this means that new incoming users must have their clock synchronized with all participants in the session in order to compute a new multicast distribution tree. Finally, in the context of a Content Delivery Network (CDN), a designated proxy Cache to serve the multimedia content is the one geographically closest to the user (even though the closest Cache not necessarily attains the minimum OWD). Having a real-time clock synchronization scheme, the CDN manager can select as the serving cache (out of several candidates) the one with the minimum OWD to the target user. This may considerably improve the service and quality of the transmitted data.

The main scenario considered in this work consists of a pair of (user or dynamic) nodes initiating a real-time multimedia session in the form of videoconference, distributed game, CDN service, application-level multicast tree, etc. We will show that under our model-based clock offset estimation scheme, it is possible to achieve good estimations in real-time rather than minutes or days as in current NTP and NTP-like schemes. It is fare to point out however, that the goal of the NTP and NTP-like protocols is to provide accurate clock synchronization over fairly long-time scales [23], while our purpose is a real-time offset estimation to satisfy the aforementioned multimedia-communication application demands.

The proposed scheme is simple, non-iterative and does not require additional overhead from communicating nodes other than timestamps. It starts working at the moment the corresponding pair of nodes exchange initial information in the way of independent $n = 5$ packet probes in the forward and reverse direction, eliminating the need of synchronous send-and-wait interaction, long sampling periods, and skew and drift corrections. Once the end-nodes have their corresponding clock offset (hereafter offset) estimations, a final offset estimation is computed by combining both independent estimates.

Finally, our scheme is non-intrusive (it works at the application level), and can be embedded into more complex multimedia distribution protocols (including the Transport Control Protocol-TCP/IP and NTP-like protocols to accelerate their convergence) for initial clock offset estimation over small periods of time. According to our knowledge, this is the first model-based proposal for clock offset estimation.

The paper is organized as follows. Basic terminology, theoretical foundation, and previous work is introduced in Section 2. Section 3 describes our proposed model for clock offset estimation, and experimental model performance and conclusions are presented in Sections 4 and 5 respectively.

2. One-way delay derivation and previous work

In this section we describe fundamental terminology for clock offset estimation and discuss previous work on offset and OWD estimation.

2.1. One-way transit time derivation

Let us define the system clock $C(t)$, as a piecewise function of t that is twice differentiable except on a finite set of points [14,15]. Then, we can state the following set of definitions regarding this function [15]:

Absolute Offset: The difference between the time reported by the system clock $C(t)$ and a "true" clock. The absolute offset of C_A is $(C_A(t) - t)$. The relative offset (or offset in our work) is then the difference between a pair of system clocks at time $t \geqslant 0$ is $\theta(t) = C_A(t) - C_B(t)$.

Frequency: The rate at which the clock progresses, $C'(t) = dC(t)/dt$.

Clock ratio or Skew (α): The frequency ratio between a clock and the "true" clock (this definition follows the terminology of [31]); the ratio of C_A is C'_B. The clock ratio of C_A relative to C_B at time t is $\alpha = C'_A(t)/C'_B(t)$.

Resolution: The minimum time unit by which the clock time is updated. It strongly depends on both the operating system and system processor. Between two hosts the reliable resolution is found in the order of tenths of milliseconds.

The relationship between two clocks can now be stated as follows:

$$C_A(t) = \theta(t) + \alpha C_B(t) \tag{1}$$

If clock information between two hosts could be exchanged instantly, we would infer the offset by using Eq. (1). However, in a communication network this is not the case and we observe additional embedded components as explained next.

A common technique for getting clock information between two hosts in a communication network is the exchange of data packets containing timing information (see Fig. 1). Let $T_A^1(i)$ denote the time when host A sends a message i to host B. Upon reception B marks this message with a new timestamp $T_B^2(i)$ and bounces it back again to A, appending a new timestamp $T_B^3(i)$ right before the packet is sent out. When host A receives the message it records the timestamp of the arrival $T_A^4(i)$. This cycle form a measurement. With the four timestamps it is easy to obtain the round trip time and one way transit time from forward and reverse paths respectively as follows:

$$RTT(i) = T_A^4(i) - T_A^1(i) \tag{2}$$

$$OWD_{A\to B}(i) = T_B^2(i) - T_A^1(i) \tag{3}$$

$$OWD_{B\to A}(i) = T_A^4(i) - T_B^3(i) \tag{4}$$

Eqs. (3) and (4) can be represented in general by:

$$OWD = T_{recv} - T_{send} \tag{5}$$

For n samples, each OWD_i, $1 \leqslant i \leqslant n$, can be split up into the following components [12]:

$$OWD_i = \theta + \xi + \delta_i + \varepsilon_i \tag{6}$$

where θ represents the offset between the clocks (assumed constant in short periods of time), ξ is the network propagation plus transmission delay, δ is the queuing delay observed by the messages or packets, and ε is a random error with zero average. For a fixed path and constant size probes, ξ is an unknown constant related to the *minimum delay* that can be observed in a path and to the physical