Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Communications 29 (2006) 1213–1225

computer communications

www.elsevier.com/locate/comcom

On estimating clock skew for one-way measurements

Jingping Bi*, Qi Wu, Zhongcheng Li

Institute of Computing Technology, Chinese Academy of Sciences, P.O. Box 2704, Beijing 100080, People's Republic of China

Received 19 September 2004; revised 18 July 2005; accepted 25 July 2005
Available online 30 August 2005

Abstract

Owning to the asymmetry of Internet paths, more and more studies have turned to the measurement of one-way metrics. Since the clocks at end systems often behave diversely, the synchronization between end hosts is what we care about all along. In this paper, we firstly propose a general model for clock skew estimation in one-way measurements, which turns the problem of clock skew estimation to the solution of n-dimension equation group, and give the equation group needed based on different presumptions. We then present Piece-wise Reliable Clock Skew Estimation Algorithm (PRCSEA), which introduces reliability test to estimation results and eliminates the extra presumptions needed by other algorithms, such as only one clock adjustment in the measurements. PRCSEA solves the skew estimation problem in a heuristic way, and it can handle many special cases affecting the estimation of clock skew, such as routing change, clock hiccup and network congestion. PRCSEA is the only algorithm that can handle clock drift to the best of our knowledge.

Keywords: Network measurement; One-way measurement; Clock synchronization; Clock skew; Drift

1. Introduction

Clock is the basis of network performance measurement. Many measurement techniques need to retrieve the time consumed by network events, such as packet-transit time in the network, server-processing time and so on. Due to the asymmetry of Internet routing [9], more and more studies have focused on the measurement of one-way metrics. Since the clocks at both end-systems are involved in delay measurement, synchronization between the two clocks becomes an important issue.

Before proceeding to detailed discussion of clock synchronization, we would like to introduce some definitions first. We define *offset* as the relative difference of the time reported by two clocks, and *skew* as the relative difference of two clocks' first order differentiation (or frequency). Furthermore, we define *drift* as the difference between two clocks' second order differentiation. Our definition consists with previous definitions in [6–8], which model computer clock as a second order differentiable function. Particularly, when we compare a clock with the *true* clock, we often omit 'relative to true clock', that is, the offset (skew, drift) of a clock is the relative (frequency, second order differentiation) difference between the clock and the true clock. Synchronization can happen at a particular moment or during a period, and the elements synchronized can be the value or frequency of clocks. We define *adjustment* as adjusting the offset to zero or under a certain level at a given moment. Furthermore, we define *time keeping* and *frequency keeping* as keeping the offset and frequency to zero or under a certain level during a period, respectively.

Although time keeping is the ideal case for clock synchronization, it is hard to achieve without the help of hardware devices like GPS or CDMA receiver. The asymmetry of network path, workload and even the bandwidth (e.g. ADSL or GPRS subscribers) makes it difficult to estimate the delay difference in two directions, which is essential for calculating clock offset [11,13]. Fortunately, in most cases frequency keeping is enough for us. For example, in delay measurement the dynamic part-mainly queuing delay-attracts much more attention than the static part composed of propagation delay and transmission delay [1,3,10,18]. Besides, many measurement methods,

---

* Corresponding author. Tel.: +86 10 62565533x9224; fax: +86 10 62567724.

E-mail addresses: jpingbi@ict.ac.cn (J. Bi), abewu@ict.ac.cn (Q. Wu), zcli@ict.ac.cn (Z. Li).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.07.023

such as available bandwidth estimation, are independent from a constant offset [16].

The key of frequency keeping is to estimate skew between two clocks. To find the essential of clock skew estimation, we propose a general model that turns the skew estimation to the solution of an *n*-dimension equation group, and use this model to evaluate existing techniques. We then present a Piece-wise Reliable Clock Skew Estimation Algorithm (PRCSEA), which introduces a test mechanism to verify the estimation results. If an estimation result in some interval does not pass the test, PRCSEA will divide the interval into smaller ones and re-estimate until the estimation results pass the test or the interval cannot be partitioned any more. To solve the clock drift problem, we propose multi-scale test (MST), which can validate the skew estimation result in log n scales, where n is the number of received packets. Combining with MST, PRCSEA is effective in the case of a long-term measurement, where the clock drift is usually non-neglectable. As far as we know, PRCSEA is the only algorithm that can handle clock drift at present time.

The rest of the paper is organized as follows. We give related works and develop the clock skew estimation model in Sections 2 and 3, respectively. In Section 4, we present the PRCSEA, together with the single-scale test and MST. We evaluate state-of-the-art algorithms using our model in many aspects and find that PRCSEA provides many good features without increasing time complexity in Section 5. The correctness and adaptability of PRCSEA is validated through measurements in Section 6. The conclusions are given in Section 7.

2. Related works

Related works can be classified by how they synchronize clocks. GPS and CDMA receivers are hardwares targeting at time keeping and the offset of clocks can be reduced to tens of microseconds with the help of them. As a result, many large network measurement projects use them to keep time in measurement probes, such as RIPE [14] and Surveyor [4]. Network Time Protocol (NTP) is a software for timekeeping and it is the most widely used time protocol in the Internet [6]. The accuracy of NTP is affected in part by the path characteristics between NTP clients and servers. The clock offset between the synchronized host and the NTP server can often be maintained on the order of multiple milliseconds [5]. However, the clock with NTP may manifest temporal clock skew or drift and these temporary clock behaviors often greatly affect the accuracy of measurement, which makes NTP not a good choice for network measurement.

Using more stable oscillators is a straightforward approach in frequency keeping and there is such a software clock based on CPU registers [12]. However, for the measurement of network internal delay [1], it is nearly impossible to install synchronization devices or upgrade software in all routers along a path. Another approach in frequency keeping is to estimate and remove clock skew from a sequence of timestamps. Paxson selects 'de-noised' One-way Transit Time (OTT) by partitioning the data into intervals and picking the minimum delay value from each interval [11], and uses them to estimate the clock skew. The approach by Moon et al is to fit a line lying under all delay points and as closely to them as possible, and use the slope of the line as the estimated skew [8]. Zhang et al compute the Convex-hull of delay points and give different skew estimations in terms of distinct objective functions [15].

There are several limitations in the above clock skew estimation algorithms. First, none of them provides verification of the estimation results. (Paxson's cumulative minima test verifies whether a continuous trend exists, not whether the estimation is correct or not.) Even though the algorithms rarely fail, the effect of the incorrect estimation may be large in many cases. Accordingly, we always need to observe and judge whether the estimations are right, which further restricts their applications. Second, the functioning of the above algorithms requires strict conditions, so Paxson and Zhang separately present techniques to detect clock adjustment, and then estimate skew to the parts without adjustments. Unfortunately, their adjustment detection algorithms also have many restrictions. Third, all of them do not take clock drift into account. Our experiments indicate that looking on the clock skew as a constant in a long-term measurement often leads to absurd conclusions.

3. Modeling clock skew estimation

To find the key of clock skew estimation and give an insight view of the existing algorithms, we develop a theoretical model with which all skew estimation techniques are comparable. Suppose node r receives a sequence of packets numbered from 1 to n from node s. Let the clocks of s and r are C_s and C_r, respectively. As in previous definitions, C_s and C_r are second order differentiable functions. We use C to denote the true clock, that is, $C(t)=t$. Let l_i^* be the moment that the i-th packet leaves s according to C_* and a_i^* be the moment that the i-th packet reaches r according to C_*, where * is the wildcard. Let d_i and D_i be the *true* delay and the measured delay of the i-th packet. Evidently, we have $d_{\mathrm{i}}=a_{\mathrm{i}}-l_i$, and $D_i=a_{\mathrm{i}}^{\mathrm{r}}-l_i^{\mathrm{s}}$. Fig. 1 shows the relation between the above variables, where the horizontal lines represent the time axes of C, C_r and C_s from the bottom upwards. The two directed diagonals from the C_s axis to the C_r axis represent the sending process of the i-th packet and the $(i+k)$-th packet, respectively. The point of intersection of the diagonal and C_s indicates the sending moment of a packet, and that of the diagonal and C_r indicates the receiving moment of a packet. D_i indicates the difference in time between the two endpoints of the diagonal.