Estimation of clock offset from one-way delay measurement on asymmetric paths

Masato TSURU
GENESIS Project*
Telecommunications Advancement Organization of Japan

Tetsuya TAKINE
Dept. of Applied Mathematics and Physics
Graduate School of Informatics, Kyoto University, Japan

Yuji OIE(*)
Dept. of Computer Science and Electronics
Kyushu Institute of Technology, Japan

Abstract

As the Internet is shifting towards a reliable QoS-aware network, accurately synchronized clocks distributed in the Internet become more significant. Network Time Protocol (NTP) is broadly deployed in the Internet for clock synchronization among distributed hosts, but it is weak in asymmetric paths, i.e., it cannot accurately estimate the clock offset between two hosts when the forward and backward paths between them have different one-way delays. In this paper, we focus on estimating the offset and the skew of a clock from one-way delay measurement between two hosts, and propose an idea for improvement of such estimations, which reduces estimation errors in case that the forward and backward paths have different bandwidths that is one of main factors of the asymmetric delays.

1. Introduction

Accurate time (clock) synchronization among distributed hosts is essential to realizing both the well-managed, QoS-aware Internet and advanced distributed applications on it. Accurately synchronized clocks on distributed hosts can play an important role in network management and research, especially in one-way packet delay measurement ([1]) that gives us valuable information on network internal states as well as end-to-end network performance. Furthermore, synchronized clocks provide not only accurate time or time-interval information but also information on the order in which distributed events occur, and thus future distributed applications will require clock synchronization in the order below milliseconds on each end-user computer.

Network Time Protocol (NTP) is broadly deployed in the Internet to synchronize distributed clocks to each other or to the time server having an accurate clock [6], [7]. Since it estimates the clock offset and skew based on one-way delay measurement across the network, the accuracy of synchronization depends on the network (the paths) between two hosts, although it can keep time within millisecond-order in various networks.

Special devices for clock synchronization, such as the Global Positioning System (GPS), are also available and can synchronize a local clock to the standard time very precisely. However, it is not ubiquitous because it requires a special environment like an antenna. Especially in the new era of "Everything on IP", since "everything" has its clock, distributed clock synchronization based on network protocols is still significant.

One of main factors increasing errors in estimate the clock offset between two hosts is the asymmetric paths problem. Since existing methods (like NTP) do not take the asymmetric paths into account, the larger difference between the one-way delay of the forward path and that of the backward path causes the more inaccuracy in the clock offset estimation.

The asymmetric paths problem arises in two typical cases. One is asymmetric bandwidths of links, and the other is asymmetric propagation delays coming from asymmetric routes (especially in distance). For a long-fat pipe (e.g., an inter-continental line), the former is negligible, but, for a short-thin pipe (e.g., an access line), the former can be dominant. Unfortunately, in the recent Internet, asymmetric bandwidths become more popular than before, such as high-speed modems, ADSLs, and cable modems. Since these lines are usually used as an access line, the end-user cannot avoid using the line. One of promising scenarios to achieve accurate (sub-millisecond) clock synchronization on end-user computers is that each ISP provides a time server near each access point. In such cases, while asymmetric distances of routes may be negligible, the problem of asymmetric bandwidths still remains.

In this paper, therefore, we focus on estimating the offset and the skew of the clock from one-way delay measurement over paths with asymmetric bandwidths. First we explain existing techniques briefly and point out the problem (Section 2), then propose an idea for improvement to reduce estimation errors in case for asymmetric bandwidths (Section 3). We also show a case study that indicates potential of our improvement (Section 4).

2. Basic model and problem with asymmetric bandwidths

2.1. Terminology

First we introduce a basic model and terminology based on [8] for describing the distributed clock synchronization problem.

Each host a has its own clock $\boldsymbol{C}_a(t)$ as a function from "true" time t to host-depended time. The "offset" is the difference between the time reported by the clock and the true time: $\boldsymbol{C}_a(t) - t$. The "frequency" is the rate at which the clock progresses: $d\boldsymbol{C}_a(t)/dt$. The "skew" is the difference between the frequencies of the clock and the true clock: $d\boldsymbol{C}_a(t)/dt - 1$. The "drift" is the rate at which the frequency varies: $d^2\boldsymbol{C}_a(t)/dt^2$.

We consider two hosts, client a and server b. For conciseness of descriptions, we assume the server's clock $\boldsymbol{C}_b$ keeps "true" time (i.e., $\boldsymbol{C}_b(t) \equiv t$), although we can evolve the almost same discussion for "relative" time synchronization without this assumption. On the other hand, the client's clock $\boldsymbol{C}_a$ may have a non-zero offset and a non-zero skew.

We also assume that conditions (e.g., temperature) of the environment in which a clock runs dose not change in a measurement term, and thus the drift is negligible so that the frequency is constant that depends on the clock. Furthermore, errors in reading a clock are negligible and the clock resolution is precise enough to make our argument meaningful.

Suppose the client and the server exchange packets in both directions (Fig. 1 and Fig. 2). For the one-way delay measurement in the direction from client a to server b (the forward path), let $\overline{c}_i$ be the timestamp of the i-th packet leaving a according to clock $\boldsymbol{C}_a$, c_i be the true time corresponding to $\overline{c}_i$ (i.e., $\boldsymbol{C}_a(c_i) = \overline{c}_i$), and s_i be the timestamp of the i-th packet arriving at b according to clock $\boldsymbol{C}_b$ (the true clock), respectively. Without loss of generality, we

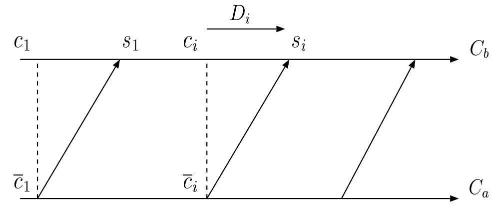

Figure 1. Timestamps of packets along the forward path (a to b)

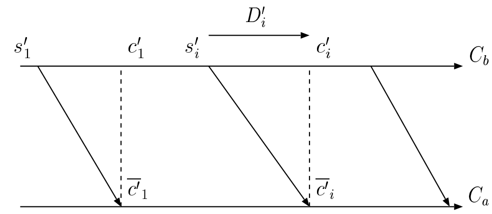

Figure 2. Timestamps of packets along the backward path (b to a)

ignore packet losses so that s_i and c_i represent a departure time and an arrival time of the same packet (the i-th packet). Similarly, for the measurement in the opposite direction (the backward path), let s'_i be the timestamp of the i-th packet leaving b according to clock $\boldsymbol{C}_b$, $\overline{c'}_i$ be the timestamp of the i-th packet arriving at a according to clock $\boldsymbol{C}_a$, and c'_i be the true time corresponding to $\overline{c'}_i$ (i.e., $\boldsymbol{C}_a(c'_i) = \overline{c'}_i$), respectively.

In what follows, we focus on techniques for estimating the offset and the skew of the client's clock from timestamp data $\{(\overline{c}_i, s_i) | 1 \le i \le N\}$ and $\{(s'_i, \overline{c'}_i) | 1 \le i \le N'\}$, which are obtained from one-way delay measurements between the client and the server.

2.2. A basic offset estimation and asymmetric paths

We show a basic technique to estimate the offset of client's clock.

First, we ignore the skew of the client's clock so that the offset is constant $o \stackrel{\text{def}}{=} \boldsymbol{C}_a(t) - t = \overline{c_i} - c_i = \overline{c'_i} - c'_i$. Suppose the client a and the server b exchange a number of packets with a constant size in both directions. Let D_i and D'_i be the one-way delays of the i-th packet along the forward and backward paths, respectively (Fig. 1 and Fig. 2).

$$\begin{aligned} D_i &= s_i - c_i - e + \varepsilon_i \\ &= s_i - \overline{c}_i + o - e + \varepsilon_i \end{aligned} \quad (1)$$

$$
\begin{aligned}
D'_i &= c'_i - s'_i - e' + \varepsilon'_i \\
&= \overline{c'}_i - s'_i - o - e' + \varepsilon'_i
\end{aligned} \quad (2)
$$

where both of e and e' represent constant errors ($e, e' > 0$) due to the difference between the timing of the clock being read for the timestamp and that of the packet leaving a sender/arriving at a receiver; both of ε_i and ε'_i are random errors with a zero average.

Then we assume the basic assumptions:

$$
\frac{1}{N}\sum_{i=1}^{N} D_i \approx \frac{1}{N'}\sum_{i=1}^{N'} D'_i \quad (3)
$$

$$
e \approx e', \quad \sum_i \varepsilon_i \approx \sum_i \varepsilon'_i
$$

From the above assumptions, we have $\sum_{i=1}^{N}(s_i - \overline{c}_i)/N + o \approx \sum_{i=1}^{N'}(\overline{c'}_i - s'_i)/N' - o$, and thus the estimated value $\hat{o}$ of o can be obtained as follows:

$$
\hat{o} \stackrel{\text{def}}{=} \frac{1}{2}\left(\sum_{i=1}^{N'}(\overline{c'}_i - s'_i)/N' - \sum_{i=1}^{N}(s_i - \overline{c}_i)/N\right)
$$

It is likely that $e \approx e'$ and $\sum_i \varepsilon_i \approx \sum_i \varepsilon'_i$. The problem is, however, assumption (3) is sometimes violated, and then the above estimation cannot work accurately. Although several improvements reduce the estimation errors due to the variation of one-way delays (e.g., queueing delays), they still assume the difference between the constant portion of the one-way delay in each direction is negligible. However, in the current Internet, asymmetric bandwidths become more popular than before, such as high-speed modems, ADSLs, and cable modems, and they may violate the above assumption and increase the estimation errors. In fact, since the bandwidth of an ADSL line is typically 512 Kbps for up-link and 1.5 Mbps for down-link, the difference between the delay time in each direction due to the asymmetric bandwidths of the line is more than 0.6 msec even for a small 64 bytes packet, which causes unacceptable errors (at least 0.3 msec in this case) for sub-millisecond clock synchronization.

2.3. A skew estimation

Next, we consider the skew (frequency) estimation. Several researches have studied the skew estimation (e.g., [6], [9], [10], [8], [11]), which are, fortunately, free from the asymmetric paths problem. We briefly show how we can estimate the skew as follows.

Let $\alpha \stackrel{\text{def}}{=} 1/(d\boldsymbol{C}_a/dt)$ and the initial offset $o \stackrel{\text{def}}{=} \overline{c}_1 - c_1$. Then we have:

$$
\begin{aligned}
\overline{c}_i - \overline{c}_1 &= \frac{1}{\alpha}(c_i - c_1) \\
c_i &= \alpha(\overline{c}_i - \overline{c}_1) + \overline{c}_1 - o
\end{aligned}
$$

Suppose the client sends a number of packets with a constant size to the server (Fig. 1). Let D_i be the one-way delay of the i-th packet along the forward path. From the above expression of c_i, we have:

$$
\begin{aligned}
D_i &= s_i - c_i - e + \varepsilon_i \\
&= s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) + o - e + \varepsilon_i
\end{aligned} \quad (4)
$$

where e represents a constant error ($e > 0$) due to the difference between the time of the clock being read for the timestamp and that of the packet leaving a sender or arriving at a receiver; ε_i is a random error with a zero average.

On the other hand, D_i consists of a constant portion d (> 0), and two variable portions q_i (≥ 0) and δ_i. q_i is the sum of queueing delays experienced by the i-th packet on each link along the path, and δ_i is the other random (error) portion with a zero average. Note that d is constant because all packets have the same size.

$$
D_i = d + q_i + \delta_i \quad (5)
$$

For conciseness of arguments, we assume the random (error) portions $\{\varepsilon_i\}$ and $\{\delta_i\}$ are small enough to be ignored. Then we have ($1 \leq i \leq N$):

$$
\begin{aligned}
q_i &= D_i - d = s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) + o - e - d \\
&= s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) - \beta
\end{aligned} \quad (6)
$$

where $\beta \stackrel{\text{def}}{=} e + d - o$.

The goal is to estimate the values of α and β from timestamp data $\{(\overline{c}_i, s_i) | 1 \leq i \leq N\}$.

Here we show a method presented in [8]. Let $Q_i(\alpha, \beta)$ denote the right-hand side of (6), that is:

$$
Q_i(x, y) \stackrel{\text{def}}{=} s_i - \overline{c}_1 - x(\overline{c}_i - \overline{c}_1) - y
$$

Because of the nature of queueing delay, we can assume that `1)` $q_i \geq 0$ ($1 \leq i \leq N$), `2)` many of q_i are likely to fall near 0, and `3)` q_i is equal to 0 for at least one i. Therefore, we estimate (α, β) so that it minimizes $\sum_i Q_i(\alpha, \beta)$ under the constraint $Q_i(\alpha, \beta) \geq 0$ for each i.

$$
(\hat{\alpha}, \hat{\beta}) \stackrel{\text{def}}{=} \arg\min_{(x,y)\in\Psi} \sum_i Q_i(x, y)
$$

where $\Psi \stackrel{\text{def}}{=} \bigcap_i \{(x, y) | Q_i(x, y) \geq 0, x > 0\}$.

In case that the above solution is not unique, we choose an appropriate candidate (e.g., the median) in $\{(x^*, y^*) \in \Psi | \sum_i Q_i(x^*, y^*) = \min_{(x,y)\in\Psi} \sum_i Q_i(x, y)\}$ as the estimated value $(\hat{\alpha}, \hat{\beta})$. It can be shown that the estimated value $(\hat{\alpha}, \hat{\beta})$ satisfies that $\min_i Q_i(\hat{\alpha}, \hat{\beta}) = 0$, so that at least one $Q_i(\hat{\alpha}, \hat{\beta})$ is equal to 0.

Although more sophisticated estimations can be considered (e.g., minimizing the weighted-sum of Q_i or using additional assumptions on the distribution of queueing delay), it was reported in [8] that the above method performs well both in an experiment in the Internet and in a simulation. An improvement for eliminating the influence of unexpected large delays due to congestion is also proposed in [11].

3. An improved offset estimation

3.1. Preparation

We propose an idea for improvement to reduce estimation errors in case for asymmetric bandwidths. Although statistical filters for measured data are necessary for eliminating exceptional errors, detecting changes of status (e.g., route alternation), and estimating unavoidable random errors, we are not concerned with them here. Suppose the client and the server exchange a number of packets with various sizes in both directions (Fig. 1 and Fig. 2). We see correlation between one-way delays of packets with different sizes. For $\boldsymbol{K} \stackrel{\text{def}}{=} \{1, 2, ..., K\}$, let $\{l_k | k \in \boldsymbol{K}\}$ be a set of packet sizes (we assume that $l_k < l_{k+1}$), L_i be the size of i-th packet, and $\boldsymbol{I}_k \stackrel{\text{def}}{=} \{i | L_i = l_k\}$ for $k \in \boldsymbol{K}$.

Furthermore, let us decompose d in (5) into some portions. J be the number of directed-links along the path, b_j be the bandwidth of the j-th link ($1 \leq j \leq J$), and $\lambda \stackrel{\text{def}}{=} \sum_j 1/b_j$. Since $d = p + f + \lambda L_i$ where $p\ (> 0)$ is the sum of signal propagation delays on each link, λL_i is the sum of transmission delays on each link, and f is the other constant portion (mainly consists of router processing delays independent of the packet size L_i), we have:

$$D_i \quad = \quad p + f + \lambda L_i + q_i + \delta_i \tag{7}$$

where q_i is the sum of queueing delays experienced by the i-th packet on each link, and δ_i is the other random (error) portion with a zero average.

Similarly, the one-way delay D'_i of the i-th packet along the backward path can be decomposed as follows.

$$D'_i \quad = \quad p' + f' + \lambda' L'_i + q'_i + \delta'_i \tag{8}$$

3.2. An idea to improve offset estimation

First we ignore the skew of the client's clock so that the offset is constant $o \stackrel{\text{def}}{=} \overline{c_i} - c_i = \overline{c'_i} - c'_i$. Then D_i and D'_i can be expressed as (1) and (2), respectively.

By ignoring the random (error) portions $\{\varepsilon_i\}$ in (1), $\{\varepsilon'_i\}$ in (2), $\{\delta_i\}$ in (7), and $\{\delta'_i\}$ in (8), we have ($1 \leq i \leq N$):

$$\begin{aligned} q_i &= s_i - \overline{c}_i - \lambda L_i - p - f + o - e \\ &= s_i - \overline{c}_i - \lambda L_i - \mu \qquad (9) \\ q'_i &= \overline{c'}_i - s'_i - \lambda' L'_i - \mu' \qquad (10) \end{aligned}$$

where

$$\mu \stackrel{\text{def}}{=} e + f + p - o, \quad \mu' \stackrel{\text{def}}{=} e' + f' + p' + o$$

If we can estimate (λ, μ) and (λ', μ') from $\{(s_i - \overline{c}_i, L_i) | 1 \leq i \leq N\}$ and $\{(\overline{c'}_i - s'_i, L'_i) | 1 \leq i \leq N'\}$, we have: $\hat{\mu} \approx e + f + p - o, \hat{\mu'} \approx e' + f' + p' + o$ where $\hat{\mu}$ and $\hat{\mu'}$ are the estimated values of μ and μ', respectively.

Moreover, if we assume: $p \approx p'$, $f \approx f'$, $e \approx e'$, then we can estimate the offset o as follows, which is our goal:

$$\hat{o} \stackrel{\text{def}}{=} \frac{\hat{\mu'} - \hat{\mu}}{2}$$

An estimation of the above unknown parameters λ and μ from measured data $\{(s_i - \overline{c}_i, L_i) | 1 \leq i \leq N\}$ originally appeared in a bandwidth estimation tool *pathcher*, and this estimation problem has been extensively studied in several researches (e.g., [2], [5], [3]). They have applied several statistical way to this kind of estimations and have discussed statistical properties such as accuracy, efficiency and robustness.

Furthermore, the same idea of the combining the clock offset estimation with the bandwidth estimation can be found in [4]. However, it does not include further details because the intention of the authors of [4] is to propose measurement methods that need to a collaboration with routers.

Here we show how we can estimate λ and μ in (9). From (9), we have ($k \in \boldsymbol{K}$):

$$\min_{i \in \boldsymbol{I}_k} q_i \quad = \quad \min_{i \in \boldsymbol{I}_k} (s_i - \overline{c}_i) - \lambda l_k - \mu \tag{11}$$

We define function $\tilde{Q}_k(x, y) \stackrel{\text{def}}{=} \min_{i \in \boldsymbol{I}_k} (s_i - \overline{c}_i) - x l_k - y$. Because of the nature of queueing delay, we can assume that 1) $\min_{i \in \boldsymbol{I}_k} q_i = \tilde{Q}_k(\lambda, \mu) \geq 0$ for each k, 2) $\tilde{Q}_k(\lambda, \mu)$ is likely to fall near 0 for each k, and 3) $\tilde{Q}_k(\lambda, \mu) = 0$ for at least one k. Therefore, we estimate (λ, μ) so that it minimizes $\sum_{k=1}^{K} \tilde{Q}_k(\lambda, \mu)$ under the constraint $\tilde{Q}_k(\lambda, \mu) \geq 0$ for each k.

$$(\hat{\lambda}, \hat{\mu}) \quad \stackrel{\text{def}}{=} \quad \arg \min_{(x,y) \in \Phi} \sum_k \tilde{Q}_k(x, y)$$

where $\Phi \stackrel{\text{def}}{=} \bigcap_k \{(x, y) | \tilde{Q}_k(x, y) \geq 0, x > 0\}$.

In case that the above solution is not unique, we choose an appropriate candidate (e.g., the median) in $\{(x^*, y^*) \in \Phi | \sum_k \tilde{Q}_k(x^*, y^*) = \min_{(x,y) \in \Phi} \sum_k \tilde{Q}_k(x, y)\}$ as the estimated value $(\hat{\lambda}, \hat{\mu})$. It can be shown that the estimated value $(\hat{\lambda}, \hat{\mu})$ satisfies that $\min_k \tilde{Q}_k(\hat{\lambda}, \hat{\mu}) = 0$.

3.3. An improved offset and skew estimation

Next, we take the existence of skew into consideration. For accurate estimations, since a relative long measurement term is necessary, the skew estimation is not negligible in general. Note that we can apply the skew estimation in the previous section using a part of data (for one direction and one size of packets) first, and then apply the previous offset estimation using the estimated skew and the whole data. However, we can show a more consistent and efficient estimation that integrates the skew estimation and the bandwidth estimation (thus the offset estimation) as follows.

Suppose the client and the server exchange a number of packets with various sizes in both directions like the above no-skew case. We consider the forward direction. From (4) and (7),

$$\begin{aligned} D_i &= s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) + o - e + \varepsilon_i \\ &= p + f + \lambda L_i + q_i + \delta_i \end{aligned}$$

By ignoring the random (error) portions $\{\varepsilon_i\}$ and $\{\delta_i\}$, we have ($1 \leq i \leq N$):

$$\begin{aligned} q_i &= s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) - \lambda L_i - p - f + o - e \\ &= s_i - \overline{c}_1 - \alpha(\overline{c}_i - \overline{c}_1) - \lambda L_i - \mu \qquad (12) \end{aligned}$$

where $\mu \stackrel{\text{def}}{=} e + f + p - o$.

We define function $\breve{Q}_i(x, y, z) \stackrel{\text{def}}{=} (s_i - \overline{c}_1 - x(\overline{c}_i - \overline{c}_1)) - yL_i - z$. Because of the nature of queueing delay, we can assume that 1) $\breve{Q}_i(\alpha, \lambda, \mu) \geq 0$ for each i, 2) many of $\breve{Q}_i(\alpha, \lambda, \mu)_i$ are likely to fall near 0, 3) $\min_{i \in \boldsymbol{I}_k} \breve{Q}_i(\alpha, \lambda, \mu)$ is likely to fall near 0 for each k, and 4) $\breve{Q}_i(\alpha, \lambda, \mu) = 0$ for at least one i. Therefore, we estimate (α, λ, μ) so that it minimizes $\sum_{i=1}^{N} \breve{Q}_i(\alpha, \lambda, \mu)$ under the constraint $\breve{Q}_i(\alpha, \lambda, \mu) \geq 0$ for each i.

$$(\hat{\alpha}, \hat{\lambda}, \hat{\mu}) \stackrel{\text{def}}{=} \arg \min_{(x,y,z) \in \Gamma} \sum_i \breve{Q}_i(x, y, z) \qquad (13)$$

where $\Gamma \stackrel{\text{def}}{=} \bigcap_i \{(x, y, z) | \breve{Q}_i(x, y, z) \geq 0, x > 0, y > 0\}$.

In case that the above solution is not unique, we choose an appropriate candidate (e.g., the median) in $\{(x^*, y^*, z^*) \in \Gamma | \sum_i \breve{Q}_i(x^*, y^*, z^*) = \min_{(x,y,z) \in \Gamma} \sum_i \breve{Q}_i(x, y, z)\}$ as the estimated value $(\hat{\alpha}, \hat{\lambda}, \hat{\mu})$. It can be shown that the estimated value $(\hat{\alpha}, \hat{\lambda}, \hat{\mu})$ satisfies that $\min_i \breve{Q}_i(\hat{\alpha}, \hat{\lambda}, \hat{\mu}) = 0$.

Similarly, we consider the backward direction. For convenience, we assume $\overline{c}_1 = \overline{c'}_1$ so that $c_1 = c'_1$ and $o = \overline{c}_1 - c_1 = \overline{c'}_1 - c'_1$. This can be realized if the server sends the first packet in the backward direction and the client sends the first packet in the forward direction just after it has received the packet from the server. Then, we have ($1 \leq i \leq N$):

$$q'_i = \alpha(\overline{c'}_i - \overline{c'}_1) - s'_i - \overline{c'}_1 - \lambda' L'_i - \mu'$$

where $\mu' \stackrel{\text{def}}{=} e' + f' + p' + o$. In the same manner as the case of the forward path, we obtain $(\hat{\alpha'}, \hat{\lambda'}, \hat{\mu'})$ as estimation of (α, λ', μ'). If these two independent measurements (and estimations) of each direction are sound (consistent), it is expected $\hat{\alpha'}$ and $\hat{\alpha}$ are very close. Thus $|\hat{\alpha'} - \hat{\alpha}|$ can indicate the soundness.

Consequently, we can estimate the offset o using $\hat{\mu}$ and $\hat{\mu'}$, under the assumptions $p \approx p'$, $f \approx f'$, and $e \approx e'$:

$$\hat{o} \stackrel{\text{def}}{=} \frac{\hat{\mu'} - \hat{\mu}}{2} \qquad (14)$$

4. A case study

4.1. An expected accuracy of the estimation

We try to analyze the accuracy of the above estimation (14). Let $\varepsilon_e \stackrel{\text{def}}{=} e' - e$, $\varepsilon_f \stackrel{\text{def}}{=} f' - f$, $\varepsilon_p \stackrel{\text{def}}{=} p' - p$, and $\varepsilon_\mu \stackrel{\text{def}}{=} \hat{\mu'} - \hat{\mu} - (\mu' - \mu)$, then we have:

$$\hat{o} - o = \frac{1}{2}(\varepsilon_e + \varepsilon_f + \varepsilon_p + \varepsilon_\mu)$$

ε_e depends on performance of two hosts, and is negligible because of its symmetry. On the other hand, ε_f, ε_p and ε_μ depend on the network (and its internal state) between two hosts. Thus, we only show a case study (an example) in which our technique is expected to be usable and useful.

Suppose a client is connected to an ISP via an access line with asymmetric bandwidths (e.g., ADSL), and synchronize its clock to the nearest time server providing by the ISP. We assume a moderate environment in which the physical conditions of the line are stable (e.g., no interference by ISDN line activities), so that the bandwidth of the line is constant. We consider whether we can estimate the clock offset o in the order below milliseconds. As mentioned in Section 2.2, existing methods not taking asymmetric bandwidths into account lead to unacceptable estimation errors. Let us investigate the magnitude of ε_f, ε_p and ε_μ in this case.

ε_f is typically affected by the difference between performance of routers along the forward and backward paths. In our case, high performance routers in the ISP back-born are negligible. Moreover, an (maybe slow) access/gateway router affects ε_f little because it is common to paths in both directions. Thus, we can ignore ε_f.

ε_p is typically affected by the difference between lengths of paths in both directions. In our case, since the time server is placed very near the access point, we also ignore ε_p.

Now let us estimate the magnitude of $\varepsilon_\mu = \hat{\mu'} - \hat{\mu} - (\mu' - \mu) = \hat{\mu'} - \mu' - (\hat{\mu} - \mu)$. For convenience, let

$$\tilde{s}_i \stackrel{\text{def}}{=} s_i - \overline{c}_1, \quad \tilde{c}_i \stackrel{\text{def}}{=} \overline{c}_i - \overline{c}_1$$

where $\tilde{s}_i$ is the receiving time of the i-th packet that ranges from $s_1 - \overline{c}_1$ to $s_N - \overline{c}_1$ and $\tilde{c}_i$ is the sending timestamp of the

i-th packet that ranges from 0 to $\overline{c}_N - \overline{c}_1$, both are relative to the sending timestamp of the first packet. The measured data can be expressed as $\mathcal{M} \stackrel{\text{def}}{=} \{(\tilde{s}_i, \tilde{c}_i, L_i) | 1 \le i \le N\}$. Let $\mathcal{D} \stackrel{\text{def}}{=} \{(c,l) | 0 \le c \le \tilde{c}_N, l_1 \le l \le l_K\}$, a rectangle region in which the projection of $\mathcal{M}$ into (c,l)-plane falls.

Since ε_μ is strongly affected by the behavior of queueing delays and how to remove the queueing delays from the measured delays, we assume the following conditions and properties.

1. The number of packets (N), the varieties of packet sizes (K) and the range of packet sizes (l_K/l_1) are large enough to ensure the following properties. For example, $N = 40000$, $K = 40$, and $l_k = 30k$ where $l_K/l_1 = 40$ and the largest packet size (1200) does not exceed the MTU size of ADSL.
2. No correlation among (unmeasurable) queueing delays $\{q_i\}$, measured timestamps $\{\tilde{c}_i\}$, and packet sizes $\{L_i\}$.
3. Concerning the behavior of small queueing delays, let $\underline{q}$ be an appropriate small value, and

$$\boldsymbol{I}_{small}(\underline{q}) \stackrel{\text{def}}{=} \{i | q_i \le \underline{q}, 1 \le i \le N\}$$
$$\mathcal{M}_{small}(\underline{q}) \stackrel{\text{def}}{=} \{(\tilde{s}_i, \tilde{c}_i, L_i) | i \in \boldsymbol{I}_{small}(\underline{q})\}$$

where the number of packets which experience queueing delays less than or equal to $\underline{q}$ is likely to be greater than 0.001 (i.e., 0.1%) of all packets. Furthermore, the projection of $\mathcal{M}_{small}(\underline{q})$ into (c,l)-plane is distributed almost uniformly on $\mathcal{D}$.

We assume we can take 0.08 ms as the above $\underline{q}$ for $N = 40000$, i.e., $|\boldsymbol{I}_{small}(0.08\,\text{ms})| \ge 0.001 \times 40000 = 40$. Later (Section 4.3) we show results of a preliminary experiment, which indicates this assumption may be feasible in a moderate environment.

Under the above assumptions, it is likely that

$$|\hat{\mu} - \mu| \le (1+\delta)\underline{q} \tag{15}$$

where δ can be 0.2.

Consequently, in our example, the accuracy of the estimation is expected to be: $|\hat{o} - o| \approx |\varepsilon_\mu|/2 \le |\hat{\mu} - \mu| \le 1.2 \times 0.08 \approx 0.096$ msec, which has a possibility of estimating the clock offset in the order below milliseconds.

4.2. Analysis of inequality (15)

We explain inequality (15). Note that δ depends on how uniformly (widely) $\mathcal{M}_{small}(\underline{q})$ and $\mathcal{M}$ are distributed on $\mathcal{D}$, and thus the greater $\underline{q}$ or N is chosen, the smaller δ can be chosen. For the true value (α, λ, μ) in (12) and the estimated value $(\hat{\alpha}, \hat{\lambda}, \hat{\mu})$ in (13), we consider the ideal plane

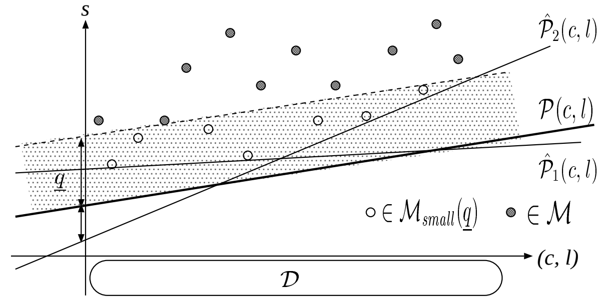

Figure 3. The ideal and estimated planes

$s = \mathcal{P}(c,l)$ and the estimated plane $s = \hat{\mathcal{P}}(c,l)$ in (s,c,l)-space, respectively (Fig. 3).

$$\mathcal{P}(c,l) \stackrel{\text{def}}{=} \alpha c + \lambda l + \mu$$
$$\hat{\mathcal{P}}(c,l) \stackrel{\text{def}}{=} \hat{\alpha} c + \hat{\lambda} l + \hat{\mu}$$

Note that 1) $\breve{Q}_i(\alpha, \lambda, \mu) = \tilde{s}_i - \mathcal{P}(\tilde{c}_i, L_i)$ and $\breve{Q}_i(\hat{\alpha}, \hat{\lambda}, \hat{\mu}) = \tilde{s}_i - \hat{\mathcal{P}}(\tilde{c}_i, L_i)$; 2) the unmeasurable data $\mathcal{M}_{small}$ must place between the ideal plane $(\mathcal{P}(c,l))$ and its parallel displacement $(\mathcal{P}(c,l) + \underline{q})$; 3) the estimated plane $(\hat{\mathcal{P}}(c,l))$ must place below $\mathcal{M}_{small}$; and 4) the projections of $\mathcal{M}_{small}$ and $\mathcal{M}$ into (c,l)-plane are distributed almost uniformly on $\mathcal{D}$, respectively.

From (13), i.e., the definition of $(\hat{\alpha}, \hat{\lambda}, \hat{\mu})$, we have $\sum_{i=1}^{N} \breve{Q}_i(\alpha, \lambda, \mu) \ge \sum_{i=1}^{N} \breve{Q}_i(\hat{\alpha}, \hat{\lambda}, \hat{\mu})$, and thus,

$$\sum_{i=1}^{N} (\hat{\mathcal{P}}(\tilde{c}_i, L_i) - \mathcal{P}(\tilde{c}_i, L_i)) \ge 0 \tag{16}$$

If two planes $\hat{\mathcal{P}}$ and $\mathcal{P}$ are parallel, then let $D \stackrel{\text{def}}{=} \hat{\mathcal{P}} - \mathcal{P} = \hat{\mu} - \mu$. From (16), we have $D \ge 0$. On the other hand, we have $D \le \underline{q}$ because of $\hat{\mathcal{P}} \le \mathcal{M}_{small} \le \mathcal{P} + \underline{q}$. Thus,

$$0 \le \hat{\mu} - \mu \le \underline{q}$$

Otherwise, $\hat{\mathcal{P}}$ and $\mathcal{P}$ have an intersection line. Let us define $\mathcal{L}$ as the projection of the intersection line into (c,l)-plane: $\mathcal{L} \stackrel{\text{def}}{=} (\hat{\alpha} - \alpha)c + (\hat{\lambda} - \lambda)l + (\hat{\mu} - \mu) = 0$. The projection of the intersection line of $\hat{\mathcal{P}}$ and $\mathcal{P} + \underline{q}$ into (c,l)-plane is $\mathcal{L} - \underline{q} = 0$. We also define $\mathcal{L}_0$ as the parallel displacement of $\mathcal{L}$ that passes through the origin $(0,0)$: $\mathcal{L}_0 \stackrel{\text{def}}{=} (\hat{\alpha} - \alpha)c + (\hat{\lambda} - \lambda)l = 0$.

Define d_i ($1 \le i \le N$), d^*, d^{**}, and h as follows (Fig. 4).

- d_i be the distance between $(\tilde{c}_i, L_i)$ and $\mathcal{L}_0$,
- d^* be the distance between $\mathcal{L}$ and $\mathcal{L}_0$,
- d^{**} be the distance between $\mathcal{L} - \underline{q}$ and $\mathcal{L}_0$, and

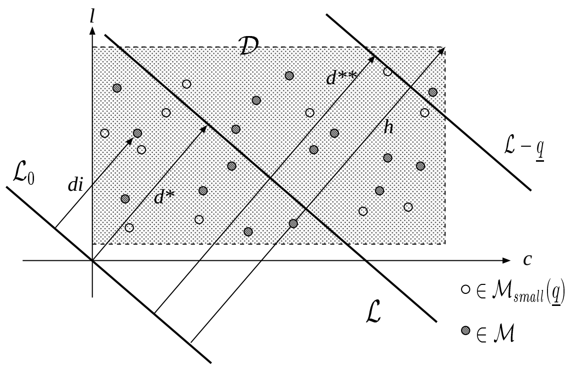

Figure 4. The projection into (c, l)-plane

- h be the distance between the farthest point from $\mathcal{L}_0$ in $\mathcal{D}$ and $\mathcal{L}_0$. In Fig. 4, the farthest point is $(\tilde{c}_N, l_K)$.

First, suppose $\hat{\mathcal{P}}(0,0) - \mathcal{P}(0,0) \leq 0$. Using the linearity of planes, we have $\hat{\mathcal{P}}(\tilde{c}_i, L_i) - \mathcal{P}(\tilde{c}_i, L_i) = C(d_i - d^*)$ for some constant $C > 0$. From (16), $C(\sum_i d_i - Nd^*) \geq 0$, and thus,

$$\sum_{i=1}^{N} d_i/N \geq d^*$$

For conciseness, we assume $\{d_i | 1 \leq i \leq N\}$ is distributed almost uniformly on $[0, h]$. Since $E[\sum_i d_i/N] = h/2$, $Var[\sum_i d_i/N] = h^2/(12N)$, and $\sum_i d_i/N$ is asymptotic normal, we have: $\Pr[d^* \leq h/2 + h/\sqrt{3N}] = \Pr[d^* \leq 0.5003h] \geq \Pr[\sum_i d_i/N \leq 0.5003h] \geq 0.95$, i.e., 95%, which implies d^* is likely to be less than $0.5003h$.

On the other hand, for conciseness, we also assume $\{d_i | i \in \boldsymbol{I}_{small}(\underline{q})\}$ is distributed almost uniformly on $[0, h]$. Recall the assumption $|\boldsymbol{I}_{small}(\underline{q})| \geq 0.001 \times N = 40$.

At least one d_i falls in $[0.93h, h]$ with the probability of $1 - (0.93)^{40} \approx 0.95$, i.e., 95%. For such i, since plane $\hat{\mathcal{P}}$ must place below $\mathcal{M}_{small}$, we have $\hat{\mathcal{P}}(\tilde{c}_i, L_i) \leq \mathcal{P}(\tilde{c}_i, L_i) + \underline{q}$. This implies d^{**} is likely to be greater than $0.93h$.

Since it is likely that $d^* \leq 0.5003h$ and $d^{**} \geq 0.93h$, using the linearity of planes, we have:

$$-1.2\underline{q} \leq \hat{\mathcal{P}}(0,0) - \mathcal{P}(0,0) = \hat{\mu} - \mu$$

Next, suppose $\hat{\mathcal{P}}(0,0) - \mathcal{P}(0,0) \geq 0$. Similar arguments can be done. From uniformity of the distributions of $\{d_i | 1 \leq i \leq N\}$ and $\{d_i | i \in \boldsymbol{I}_{small}(\underline{q})\}$, it is likely that $d^* \geq 0.4997h$ and $d^{**} \leq 0.07h$, respectively. Hence, we have:

$$\hat{\mu} - \mu = \hat{\mathcal{P}}(0,0) - \mathcal{P}(0,0) \leq 1.2\underline{q}$$

4.3. A preliminary experiment

We sent a number of pings over an ADSL line, and measured their RTT (round trip time) to a server or a router near the access point. We performed 10000 pings in the rate of 10 packets per second each two hours in each week day. To see the behavior of (small) one-way queueing delays, we observed $Q_i \stackrel{\text{def}}{=} (\text{RTT}_i - \min_i \text{RTT}_i)/2$ for $1 \leq i \leq 10000$, which can be regarded as an approximate queueing delay in this preliminary experiment.

We selected the best (stable and sound) five measurements from the twelve measurements in each day. Table. 1 shows the summary of such five measurements in a day (and five days in a week) using the following notations. A: $\min_i \text{RTT}_i/2$ (ms), B: the number of packets that may experience queueing delays less than or equal to $\underline{q}$, i.e., $Q_i \leq \underline{q}$, in all 10000 packets. As $\underline{q}$, we choose 0.08 ms for the ADSL line.

In such simple measurements, variations of the RTT measured by the ping command may include much uncertainty apart from queueing delays, due to the local system or the responder system. However, we can see a rough tendency of the behavior of small queueing delays, and we can expect that the number of packets that experience queueing delays less than or equal to 0.08 ms is likely to be greater than 10 (i.e., 0.1% of all packets) under moderate conditions.

Table 1. The small delays over an ADSL line

day	A(max/avg/min)			B(max/avg/min)		
`Mon`	19.08	19.07	19.06	17	13.2	9
`Tue`	19.06	19.05	19.04	14	11.0	9
`Wed`	19.08	19.07	19.05	16	10.4	8
`Thu`	19.07	19.06	19.05	13	11.0	9
`Fri`	19.07	19.06	19.05	17	12.6	7
avg.	19.072	19.062	19.050	15.4	11.6	8.4

5. Concluding Remarks

In this paper, we have presented an idea for improvement of the clock offset estimation, which reduces estimation errors in case that the forward and backward paths have different bandwidths. For an ADSL line with 1.5 Mbps/512 Kbps bandwidths, for example, the 0.3 msec error due to asymmetric bandwidths is expected to be reduced to less than 0.1 msec.

To realize the accurate clock synchronization (submillisecond) in the Internet, much work remains to be done. More analysis, simulations or experiments are necessary in various scenarios, which clarifies the reliability and limitations of our idea. Furthermore, we have not been concerned with important issues, e.g., statistical filtering / validations, local clock adjustments, and scalable and robust protocols / algorithms to be implemented (as an improvement of NTP), which are essential to develop a practical method.

Another asymmetry with respect to propagation delays is still a big issue for the accurate clock synchronization in the Internet, especially in satellite links, for example. Methods to detect whether asymmetric delays exist or not, and moreover, to distinguish asymmetric bandwidths and asymmetric propagation delays are also important.

Acknowledgment

We thank the QGPOP project (`www.qgpop.net`) for providing us with RTT data on an ADSL line. We also thank reviewers who give us valuable suggestions for this paper.

References

[1] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way delay metric for ippm. RFC 2679, Network Working Group, Sept. 1999.

[2] A. B. Downey. Using pathchar to estimate internet link characteristics. In *Proc. ACM SIGCOMM*, pages 241–250, Massachusetts, Aug 1999.

[3] W. Jiang and T. F. Williams. Detecting and measurement asymmetric links in an ip network. In *Proc. IEEE Globecom*, Rio de Janeiro, Brazil, Dec 1999.

[4] D. D. Luong and J. Biro. Needed service for network performance evaluation. In *Proc. IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, England, July 2000.

[5] K. Matoba, S. Ata, and M. Murata. Improving bandwidth estimation for internet links by statistical methods. *IEICE Trans. Commun.*, E84-B(6):1521–1530, June 2001.

[6] D. L. Mills. Network time protocol (version 3) specification, implementation and analysis. RFC 1305, Network Working Group, Mar. 1992.

[7] D. L. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Trans. Networks*, 3(3):245–254, June 1995.

[8] S. B. Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *Proc. IEEE infocom*, New York, Mar 1999.

[9] V. Paxon. *Measurements and analysis of end-to-end Internet dynamics*. PhD thesis, University of California, Berkely, 1997.

[10] V. Paxon. On calibrating measurements of packet transit times. In *Proc. ACM SIGMETRICS*, Madison, June 1997.

[11] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda. Achieving moderate fairness for udp flows by path-status classification. In *Proc. IEEE LCN*, pages 252–261, Nov 2000.