International Journal of Advanced Science and Technology
Vol. 35, October, 2011

Survey on the Clock Synchronization Schemes for Propagation Delay Measurement

Minsu Shin[1], Mankyu Park[1], Deockgil Oh[1], Byungchul Kim[2] and Jaeyong Lee[2]

[1]*Dept. of Satellite Wireless Convergence,*
Electronics and Telecommunications Research Institute, Korea
{msshin, neomkpark, dgoh}@etri.re.kr
[2]*Dept. of Information Communication Engineering,*
Chungnam National University, Korea
{byckim, jyl}@cnu.ac.kr

Abstract

In this paper we present the comprehensive survey on the clock synchronization algorithms, which should be considered for the measurements of the network delay. We categorize the clock synchronization algorithms into two basic types according to how they acquire synchronization between clocks, which are external source based schemes and end-to-end measurement based schemes. While external source based schemes are the synchronization methods using centralized time source such as NTP, GPS and IEEE 1588 to have global synchronization for all end hosts, end-to-end schemes obtain synchronization information through network measurements between end hosts. We briefly introduces some algorithms in both categories. However, we have focused more on the end-to-end schemes which can be classified again into online and offline shemes according to whether they can be applied for real time operation. We survey the recent progresses on these end-to-end algorithms and special concerns are on the estimation of true the one-way delay without the effect of clock skew. The problems in depolying each end-to-end scheme are also described. The potential further research issues in online one-way delay estimation are discussed.

Keywords :*clock synchronization, delay measurement, one-way delay, clock skew*

1 Introduction

The fast expansion of the Internet to deliver increasingly important and various services make network monitoring and performance measurements essential for effective network management. Many applications may benefit from the knowledge of the end-to-end delay metrics. Network latency is an important indicator of the operating network status, which changes with the variations of the network traffic patterns and congestions. Many QoS sensitive applications require the delay constraints to be met. Therefore, the knowledge of the end-to-end delay can be used for Service Level Agreement(SLA) validation between network service providers and customers. Through the end-to-end delay measurements, researchers can learn more about the underlying properties or characteristics of the current networks, for example, network topology, traffic patterns and protocol distributions, etc.

In addition, end-to-end delay metrics are widely utilized to the algorithms for performance enhancement of some protocol including TCP since they are the foundations for many other measurement metrics such as bandwidth, jitter and packet loss. While Round Trip Time (RTT) is basically used to represent end-to-end delay, the needs for One-Way Delay (OWD) measurement are also addressed due to some reasons. Measuring the one-way delay instead of the round-trip delay is motivated by reasons, such as asymmetricities of path and queuing as well as each application's characteristics[3]. In real measurements, the delay shows changing trend of about 100 msec over the duration of 70 min because of the clock difference[4].

- Because in current Internet, there may exist asymmetric paths, which means that the performance of forward and reverse path between two end hosts may be different.
- Even when the two paths are symmetric, the different and asymmetric queuing in the routers may result in different performance characteristics.
- In addition, some applications such as FTP, Video on Demand (VoD) are more dependent on the performance in one direction.

End-to-end one-way delay experienced by a packet is the time taken to travel from source to destination by a packet and can be measured from the difference between the arrival time, according to the destination clock, and the timestamp added by the source and conveyed by the packet.

$$t^{s}_{i,OWD} = t^{r.a}_{i} - t^{s.l}_{i} \ . \tag{1}$$

If the two clocks at the both end hosts are perfectly synchronized, then the one-way delay can be calculated by subtracting the sender timestamp from the receiver timestamp and this measured delay will be the true delay between the two end hosts. However, two clocks are rarely perfectly synchronized in real systems. The clocks may have different values at a certain moment and they may run at different speeds. Since the clocks at both end hosts are involved in delay measurement, synchronization between two clocks becomes an important issue.

Before proceeding the discussion on one-way delay measurements, we would like to introduce some terminologies for the clock behavior which are usually accepted in literaure[4]. The relative difference of the time reported by two clocks is called *offset*, the rate at which the clock progresses is called *frequency* and the relative difference of two clocks's frequency is called *skew*. In addition, it is sometimes convenient to compare the frequency ratio between two clocks instead of the skew. This is called *clock ratio*. Due to the offset and skew between two clocks, end-to-end delay measurements becomes inaccurate and the expected performance enhancement from the measurement results is not quaranteed.

To solve the clock synchronization problems, many algorithms and methods are introduced. There are two kinds of clock synchronization approaches according to how they synchronize clocks : external server based methods and end-to-end measurement based methods. Basic idea of external server based methods is to locate global server providing time information to every host in the network. Every host have to recognize the server and operate under the time synchronization through the server. Network Time Protocol(NTP)[6], Global Positioning System(GPS) and IEEE 1588[7] could be included within this category. While external server based methods focuses on the synchronization of the clock itself, because that is the first uncertainty source, end-to-end measurement

Table 1. Taxonomy of the Clock Skew Removal Schemes.

No.	Schemes	on/offline
1	Average technique	offline
2	Direct skew removal technique	offline
3	Sliding window algorithm	online
4	Combined algorithm	online
5	PRCSEA	offline
6	ME based	offline
7	Sync-TCP	online
8	Likelihood detector	online
9	Hybrid method for NCS	online
10	D. Kim	online
11	J. H. Choi	online
12	Tsuru	offline
13	S. Moon	offline
14	Paxsonn	offline
15	ESRS	online
16	J. Wang	offline
17	RTSE	online
18	Z. Yang	online
19	Convex hull	off/online
20	C. Guang	offline
21	M.Aoki	offline

based methods mainly focus on the detection and removal of the clock skew existing between two clocks so that time values generated by each of the end hosts can be considered to be synchronized.

In this paper, we survey the time synchronization issues in recent years, especially for the one-way delay measurements. The rest of the paper is organized as follows. We define the basic one-way delay model and notations used through the paper in Sec. 2, and describe the brief review for the external server based methods in Sec. 3. In Sec. 4, we focus intensively on the end-to-end measurement based methods. End-to-end methods are classified into offline and online schemes. In Sec. 5, we discuss issues to be considered for such an algorithms proposed for real time operation to be incorporated with the transport protocols. Lastly, we conclude this paper in Sec. 6.

2 Basic Terminology and One-Way Delay Model

To understand the effect of clock skew and the insight idea for methods to remove it, we need to define the relation between time instances and corresponding delays. In this section we introduce the terminology for clocks, timestamps, and delays used in measurements. To get consistent with previous works, we mainly adopt following nomenclature from [4][12] to characterize clock.

In Fig. 1, C_s and C_r denote sender and receiver clocks. For all values, superscripts s and

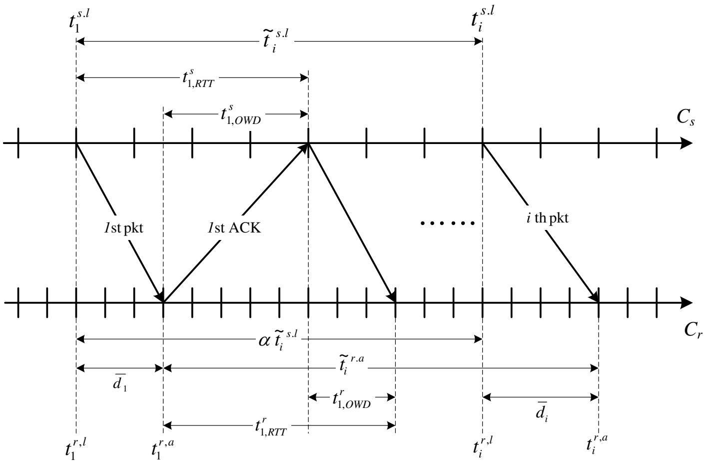

Figure 1. Relation between Timing Information.

c means that the value is measured according to the sender clock C_s and receiver clock C_r, respectively, and another superscripts a and l means the arriving and leaving timestamps, respectively. Moreover, $\bar{d}_i$ is end-to-end delay consistent with the true clock.

- C_s, C_r : sender and receiver clock.
- $t_i^{s.l}$: timestamp of the i-th packet just before its leaving according to C_s.
- $t_i^{r.l}$: timestamp of the i-th packet just before its leaving according to C_r.
- $t_i^{s.a}$: timestamp of the i-th probe arriving at the receiver according to C_s.
- $t_i^{r.a}$: timestamp of the i-th probe arriving at the receiver according to C_r.
- α : the clock ratio between C_s and C_r.
- $t_{i,RTT}^{s}$: round trip time for i-th packet measured at the sender.
- $t_{i,RTT}^{r}$: round trip time for i-th ACK packet measured at the receiver.
- $t_{i,OWD}^{s}$: one-way delay for i-th packet measured at the sender.
- $t_{i,OWD}^{r}$: one-way delay for i-th ACK packet measured at the receiver.
- $\tilde{t}_i^{s.l}$: the time duration between the first and i-th packets' leaving time according to C_s.
- $\tilde{t}_i^{r.a}$: the time duration between the first and i-th packets' arriving time according to C_r.
- $\tilde{d}_i$: relative delay calculated from $t_i^{s.l}$ and $t_i^{r.a}$.
- $\bar{d}_i$: end-to-end delay consistent with the true clock.

Fig. 1 shows the timing relation between C_s and C_r when C_s and C_r run at different frequency. The end-to-end delay of the i-th packet consistent with C_r is $t_i^{r.a}$ - $t_i^{r.l}$. However $t_i^{r.l}$ is not known at the receiver, so $t_i^{r.a}$ - $t_i^{s.l}$ is typically used as one-way delay, which is not consistent with neither C_s nor C_r. To make it consistent with either clock, we need to determine the skew between two clocks and remove it from the observable one-way delay.

3 External Server based Synchronization

Several external mechanisms have been introduced to physically synchronize the end host's clocks so that the time information from both of the end hosts have no offset and skew. The Network Time Protocol(NTP), Global Positioning System (GPS) and IEEE 1588[7] can be included within the categoriy.

NTP is broadly deployed in the Internet to synchronize distributed clocks to each other or to the time server having an accurate clock. It can provide accuracy typically within a millisecond on LANs and up to a few tens of milliseconds on WANs[8]. The NTP system consists in a hierarchy of primary and secondary time servers, clients and interconnecting transmission paths[9]. Under normal circumstances clock synchronisation is determined using only the most accurate and reliable servers and transmission paths, so that the actual synchronisation paths usually assume a hierarchical configuration with the primary reference source at the root and servers of decreasing accuracy at increasing layers toward the leaves. Although the clock offset between the synchronized host and the NTP server can often be maintained on the order of multiple milliseconds, the accuracy of NTP is affected in part by the path characteristics between NTP clients and servers, which makes NTP not a good choice for an accurate network measurement.

GPS system can also be used for the clock syncronization with high accuracy in the order of microseconds and wide coverage for large scale networks. Many measurement architectures incorporated with the GPS system have been proposed[8][10]. However, it requires additional hardware systems such as antenna and distribution equipment for every or group of hosts, which make its use impractical in the viewpoint of economy and convienience.

The new IEEE Standard Precision Time Protocol (PTP), IEEE1588[7], is now a very comprehensive solution to do very precise time synchronization in an Ethernet network. The IEEE 1588 time synchronization protocol specifes how such synchronization can be achieved over wired and wireless networks. It is an external synchronization protocol in which all clocks in the network trace their time. This protocol is the first standard available which makes it possible to synchronize the clocks of different end devices over a network at speeds faster than one microsecond. The protocol supports system-wide synchronization accuracy in the submicrosecond range with minimal network and local clock computing resources. However, its accuracy can be affected from the network fluctuations introduced by network elements such as switches and routes, which imposes limitation of its use to small network.

Aforementioned external server based methods for the time synchronization have their own advantages and potentials for further considerations. However, they are, at least at the moment, not appropriate solution due to their practical limitations in terms of the cost and the susceptibility to network characteristics.

4 End-to-end measurement based Synchronization

Although time keeping is the ideal case for clock synchronization, it is hard to achieve without the help of hardware devices like GPS or hardware-based NTP server as described in Sec. 3. Another approach for time synchronization is to find clock uncertainty existing between sender and receiver clocks and remove it to make both clocks almost perfectly synchronized. Most alogorithms included in this category perform the estimation and removal process of clock skew from the network measurement while external server based methods try not to have any uncertainty in clock dynamics through providing the same clock source to every end host in the network. However, the asymmetry of the network path, the amount of traffic flow and the bandwidth makes it difficult to estimate the delay difference in two directions, which is essential for calculating clock offset between two hosts. Fortunately, in most cases frequency keeping is enough for this purpose. 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. Besides, many measurement methods, such as available bandwidth estimation, are independent from a constant offset. From these observations, many contributions are devoted to determine the clock skew in the measurement and by removing the effect of the skew we can transform the delay measurements so that they are consistent with a single clock. In some cases, one-way delay metrics are more important than round-trip time measurement with the reasons exampled in Sec. 1. Therefore, most of algorithms have focused on the detection and estimation of clock skew existing in the unidirectional path.

End-to-end measurement based synchronization methods can be classified into two subgroups according to their real-time applicability, i.e., offline syncronization method and online synchronization method. Offline synchronization methods carry out their estimation and removal of clock skew with the prerequisite of network measurements taken for a certain period of time ahead. On the other hand, online synchronization methods calculate clock skew immediately on receiving packets involved for the calculation and derive required metrics through the clock skew removal. Proposed schemes in the literature are described in the following two sections, respectively.

4.1 Offline Synchronization Approches

Offline synchronization methods calculate the clock skew existing between sender and receiver clocks from the network measurement data.

To deal with the clock synchronization problems such as relative offset and skew, Paxson propsed the median line fitting algorithm using forward and reverse path measurements of delay between a pair of hosts[11][12]. Moon *et al.* focused on filtering out the effects of clock skew only with the unidirectional delay measurement to determine the variable portion of the delay. The basic idea of the algorithm is to fit a line lying under all delay points and being as closely to them as possible, and use the slope of the line as the estimated skew[4]. The authors formulated this idea as a linear programming problem.

Khlifi *et al.* proposed two offline clock skew estimation and removal algorithms[13]. They formalized the clock skew model in Eq. (2), which is identical to the model in [4].

$$d_i = d_i^r + (\alpha - 1)t_i^s + \theta \ . \tag{2}$$

, where d_i is the measured delay experienced by the i-th packet, d_i^r is the ture one-way delay experienced by i-th packet, α is the clock skew between two clocks, t_i^s is defined by the difference of generation times for i-th and the first packets according to the sender clock, and θ is the relative offset between two clocks. In Average technique, they adopted the notion of the phase plot to show the evolution of the difference between the packe delays

$d_i - d_{i-1}$ and thus provided the way of calculating the estimate of the skew, $\widehat{\alpha}$ only with the obtainable values at the very low complexity in eq. (3).

$$\widehat{\alpha} = \left(\frac{d_k - d_{l-1}}{k-l-1}\right) / \left(\frac{t^s_k - t^s_{l-1}}{k-l-1}\right) + 1 \ . \tag{3}$$

, where k and l are the indexes of the minimun measured delay for the two same sized intervals selected from the begining and end of measurement trace and considered to avoid the possible extreamities in the boundaries of the trace. Another algorithm proposed by khlifi *et al.*, which is called direct skew removal technique, is to estimate the true one-way delay directly under the assumption that the minimum system clock resolution is equal to 1 msec and thus the variation in measured delay due to the skew is increased or decreased by the multiple of 1 msec, in the form of steps, depending on the sign of the skew (e.g., if the sender clock progresses faster than receiver clock, then the skew becomes negative). They have also considered the effect in the presence of clock resets.

A quite different approache called Piece-wise Reliable Clock Skew Estimation Algorithm (PRCSEA) from the other previous algorithms was presented by Bi *et al.* in that it provides reliablility test to estimation results so that evenfually it doesn't care about the presumption for clock dynamics[5]. Most algorithms in the field do not handle clock drift, assuming that the clock skew ramains constant, because it is very hard to decide where the skew changes in reality. PRCSEA takes those clock dynamics into consideration and introduces verification to estimation results so that it can handle the clock drift by naturally eliminating the needs to identify the skew changing point in its recursive process. Instead of providing specific skew estimation algorithm, it focuses on verifying the results of skew estimation using any existing algorithm for that. The authors showed that it has low time compexity even when there exits clock adjustment and drift within the measurement by evaluation and it performs well across diverse clock dynamics by simulation.

4.2 Online Synchronization Approches

Online end-to-end synchronization schemes are aiming at the estimation and removal of the clock skew in the real-time manner upon receiving packets and thus can be utilized to improve the performance of the operating protocols, e.g., it can be adopted for the bandwidth estimation or loss differenciation shemes to enhance TCP congestion control performance. However, not much work has been done for online skew estimation and removal so far comparing to the offline algorithms.

Tobe *et al.* presented a simple scheme for estimating the skew through reducing the number of samples for calculation, called Estimation of Skew with Reduced Samples (ESRS) to alleviate the problem of collecting many samples for long period, which makes an algorithm unsuitable for on-line calculation of the skew[14]. While adopting similar delay model in [4], they proposed some modifications. To reduce the number of calculation, a measurement that has an inter-arrival time outside the expected range is not taken into consideration from the observation that the variable inter-arrival time induced by the network characterisics such as probe compression should be excluded in the skew calculation. With this reduction process, it was shown that the significant portion of samples can be removed depending on the network status, e.g., 86% of reduction factor can be obtained when the network seems to be unstable or fluctuated with other traffics. Moreover, the algorithm enhances the estimate of the base delay incrementally on arrival of packets rather than calculating the skew after a certain period so that the estimate of the skew converges to a certain value and the OTT value without the clock skew can be determined thereafter. This skew estimation scheme was utilized for loss differentiation algorithm in [15] by which that the detected packet loss is determined as congestion loss in TCP congestion control loop when the Relative One-Way Trip Time (ROTT) value calculated by ESRS has been maintained over a threshold.

Khlifi *et al.* proposed sliding window technique and combined approach for the online synchronization[13]. The basic idea of the sliding window technique is to continuously evaluate the variation of the minimum measured delay. For this, the length T of the evaluation interval called window size is chosen and the minimum measured delay is determined for every interval. In the case that the minimum delay of the current inveral is smaller than the one of the previous interval, the algorithm decides that the skew is negative and the skew value is decreased by 1. In the opposite case, the algorithm increases the skew value by 1. Then the true delay can be obtained by the every measured delay minus this skew value. This technique gives quick response to skew effect with good accuracy so that it could be applied for online synchronization. However, the choice of the window size, T, has great influence on the performance of the algorithm and the window size has to be the duration that the skew effect takes to reach 1 *ms* to quarantee its performance. Since the proper window size is unlikely to know in reality and totally depedns on the clock resoltion of the end systems, it could be limited to be widely used for online applications.

In [16], the authors propose to use their convex hull based algorithm (originally designed for offline skew estimation) to remove the skew from online delay measurements. Their idea is to estimate the skew at fixed intervals and to use the last estimate to remove the skew effect from upcoming measures.

In [18], Choi *et. al* proposed one-way delay estimation algorithm without clock synchronization between sender and receiver clocks. They made analytical derivation of forward and reverse delay separately in terms of two RTT values measured by the sender and the receiver.

$$\begin{aligned} t^s_{n,OWD} &= t^s_{0,OWD} - \sum_{i=1}^{n} \, [t^s_{i,RTT} - t^r_{i,RTT}] \\ t^r_{n,OWD} &= -t^s_{0,OWD} + \sum_{i=1}^{n} \, [t^s_{i,RTT} - t^r_{i,RTT}] + t^s_{n+1,RTT} \ . \end{aligned} \tag{4}$$

As seen in Eq. (4), they did not focus on the detection of clock skew and utilized the fact that the time duration between two leaving or arriving events is not dependent on the presence of the skew. This idea leads to estimate one-way delay just with the obtainable measures upon receiving each packet such as RTT values at the sender and receiver. With this characteristic, the algorithm can be incorporated with any end-to-end tranport protocol and the performance enhancement for TCP with this one-way delay estimation algorithm had been shown in [18]. However, some uncertainty can be induced by the heuristics for determining the initial paramter, i.e., $t^s_{0,OWD}$ in Eq. (4), and overall accuracy of the algorithm can be affected by this ambiguity.

As another approach using one-way delays in forward and reverse paths which is similar with [18], Kim *et al.* presented end-to-end one-way delay estimation sheme using one-way delay variation and round-trip time (RTT)[19]. This algorithm is based on the idea that one-way delay variation, i.e., jitter, depends only on the difference of RTTs because the effects of clock skew can be naturally removed in the RTT calculation process. They showed mathematically that the jitters for each direction can be given using RTTs measured at the sender and the receiver without *a priori* clock synchronization, and furthermore the ratio of one-way delays are equal to the ratio of one-way jitters expressed by the measured RTT assuming a certain condition. By the way that the measurements for RTTs and estimations of one-way delays and offsets are made only with the samples satisfying the condition, they, eventually, determines the unknown one-way delays from the obtainable RTT values with the reduced calculations. With these processes, the algorithm can provide following characteristics : without any assumption of time synchronization, it can track the variations of one-way delays in real time and work well under realistic network conditions because it just takes samples satisfying the certain condition.

5 Open Issues in Online OWD Measurements

Although many researchers have devoted their efforts on the clock synchronization algorithms for the one-way delay measurement and therefore many valuable contributions are presented in the literature, Online delay calcualtion algorithms are not much than the offline algorithms. Moreover, even such an online calculation algorithms mentioned in Sec. 1 and Sec. 4.2 are rarely adopted to the operation of any transport protocol in reality, to enhance their performance for example. We want to discuss the reasons in the paper.

First of all, TCP, as one of typical transport protocols in the internet, is a sender oriented protocol which means that it is prefered for every modification in the protocol to be involved only to the sender side. This is required to quarantee that all end hosts do not have to be changed. However, for calculation and delivery of the forward one-way delay measurement which is more important for TCP congestion control than reverse one-way delay, receiver side modification is inevitably required. To address this issue, the way of modifying the current TCP timestamp option can considered as in Sync-TCP [17].

6 Conclusion

In this paper we present the comprehensive survey on the clock synchronization algorithms in recent years, which should be considered for the measurements of the network delay. By defining the end-to-end delay model, we analyze the important factors and their processes of the previous works in a unified way. We categorize the clock synchronization algorithms into two basic types according to how they acquire synchronization between clocks, external source based schemes and end-to-end measurement based schemes. While external source based schemes are the synchronization methods using centralized time source such as NTP, GPS and IEEE 1588 to have global synchronization for all end hosts, end-to-end schemes obtain synchronization information through network measurements between end hosts. We briefly introduces some of algorithms in both categories. However, we have focused more on the end-to-end schemes which can be subdivided into online and offline shemes according to whether they can be applied for real time operation. We survey the recent progresses on these end-to-end algorithms and special concerns are on the clock synchronization for the one-way delay measurements. The problems in depolying each end-to-end scheme are also described. As network bandwidth increases dramatically and the asymmetry becomes more likely, inaccurate measurements for the network characteristics will cause potential network performance degradation. In that sense, the consideration for these clock dynamics should be prudently taken with the shemes summarized in the paper.

Acknowledgments.

This work was supported by the IT R&D program of KCC[2009-S-039-02, Development of Satellite Return Link Access Core Technology for High Efficient Transmission].

This paper is a revised and expanded version of a paper entitled "Clock Synchronization for One-Way Delay Measurement : A Survey" presented at the 3rd International Conference on Advanced Communication and Networking (ACN2011), Brno University of Technology, Czech Republic, 15–17 August 2011.

References

[1] J. C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," *Journal of High-Speed Networks*, vol. 2, no. 3, pp. 305–323, Dec. 1993.

[2] D. L. Mills, "Improved algorithms for synchronizing computer network clocks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 245–254, Jun. 1995.

[3] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-Way Delay Metric for IPPM," *IETF RFC 2679*, Sep. 1999.

[4] S. Moon, P. Skelley, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. IEEE INFOCOM 1999*, New York, NY, Mar. 1999.

[5] J. Bi, Q. Wu, and Z. Li, "On estimating clock skew for one-way measurements," *Computer Communications*, pp. 1213–1225, 2006

[6] D. L. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis," *IETF RFC 1305*, 1992.

[7] IEEE Std. 1588-2088, "IEEE Standard for a Precision Clock Synchronization protocol for Networked Measurement and Control Systems," *IEEE*, Jul. 2008.

[8] L. D. Vito, S. Rapuano, and L. Tomaciello, "One-Way Delay Measurement : State of the Art," *IEEE Trans. Instrumentation and Measurement*, vol. 57, no. 12, pp. 2742–2750, Dec. 2008.

[9] A. S. Sethi, H. Gao, D. L. Mills, "Management of the Network Time Protocol(NTP) with SNMP," Technical Report No. 98-09, Nov. 1997.

[10] J. Jeong, S. Lee, Y. Kim, and Y. Choi, "Design and Implementation of One-Way IP Performance Measurement Tool," Springer-Verlag, vol. 2343, no. 2, pp. 673–686, London, U.K., Jun. 2002, .

[11] V. Paxson, "On calibrating measurements of packet transit times," in *Proc. ACM SIGMETRICS 1998*, Madison, WI, Jun. 1998, pp. 11–21.

[12] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. dissertation, Lawrence Berkeley Nat. Lab., Univ. California, Berkeley, 1997.

[13] H. Khlifi and J. C. Grgoirea, "Low-complexity offline and online clock skew estimation and removal," *Computer Networks*, vol. 50, issue 11, pp. 1872–1884, Aug. 2006.

[14] Y. Tobe, H. Aida, Y. Tamura, and H. Tokuda, "Detection of change in one-way delay for analyzing the path status," in *Proc. of the Passive and Active Measurement Workshop (PAM2000)*, pp. 61–68, Apr. 2000.

[15] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, "Achieving moderate fairness for UDP flows by path-status classification," in *Proc. 25th Annu. IEEE Conf. Local Computer Networks (LCN 2000)*, Tampa, FL, Nov. 2000, pp. 252–261.

[16] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," in *Proc. IEEE INFOCOM 2002*, New York, NY, Jun. 2002, pp. 160–169.

[17] M. C. Weigle, K. Jeffay, and D. Smith, "Delay-based early congestion detection and adaptation in TCP : impact on web performance," *Computer Communications*, pp. 837–850, 2005

[18] J. H. Choi and C. Yoo, "One-way delay estimation and its application," *Computer Communications*, pp. 819–828, 2005

[19] D. Kim and J. Lee, "One-way delay estimation without clock synchronization," *IEICE Electronics Express*, vol. 4, no. 23, pp. 717–723, Dec. 2007.

[20] M. Aoki, E. Oki, and R. Rojas-Cessa, "Measurement Scheme for One-Way Delay Variation with Detection and Removal of Clock Skew," *ETRI Journal*, vol. 32, no. 6, pp. 854–862, Dec. 2010.