Measurement Scheme for One-Way Delay Variation with Detection and Removal of Clock Skew

Makoto Aoki, Eiji Oki, and Roberto Rojas-Cessa

One-way delay variation (OWDV) has become increasingly of interest to researchers as a way to evaluate network state and service quality, especially for real-time and streaming services such as voice-over-Internet-protocol (VoIP) and video. Many schemes for OWDV measurement require clock synchronization through the global-positioning system (GPS) or network time protocol. In clock-synchronized approaches, the accuracy of OWDV measurement depends on the accuracy of the clock synchronization. GPS provides highly accurate clock synchronization. However, the deployment of GPS on legacy network equipment might be slow and costly. This paper proposes a method for measuring OWDV that dispenses with clock synchronization. The clock synchronization problem is mainly caused by clock skew. The proposed approach is based on the measurement of inter-packet delay and accumulated OWDV. This paper shows the performance of the proposed scheme via simulations and through experiments in a VoIP network. The presented simulation and measurement results indicate that clock skew can be efficiently measured and removed and that OWDV can be measured without requiring clock synchronization.

Keywords: Delay measurement, clock skew, clock rate adjustment, one-way delay, inter-packet delay.

Manuscript received Oct. 27, 2009; revised June 25, 2010; accepted Aug. 23, 2010.

This work was partly supported by the Invitation Fellowship Program for Research in Japan of Japan Society for the Promotion of Science (JSPS).

Makoto Aoki (phone: +81 3 5459 3185, email: aoki@cybersoken.com) is with the Cyber Creative Institute, Tokyo, Japan, and also with the University of Electro-Communications, Tokyo, Japan.

Eiji Oki (email: oki@uec.ac.jp) is with the Department of Information and Communication Engineering, University of Electro-Communications, Tokyo, Japan.

Roberto Rojas-Cessa (email: rojas@njit.edu) is with the Department of Electrical and Computer Enginnering, New Jersey Institute of Technology, Newark, NJ, USA.

doi:10.4218/etrij.10.0109.0611

I. Introduction

Real-time applications over the Internet, such as voice-over-Internet protocol (VoIP) and video streaming services, use one-way delay variation (OWDV) information to monitor and maintain quality of service (QoS) or service quality as perceived by the user [1]. One-way delay (OWD) is defined as the delay that a data packet suffers in the trip from source to destination. If the source and destination use the same clock, this is equivalent to the difference between the arrival time at the destination and the departure time at the source. OWDV is defined as the difference in OWD experienced by two packets of the same length [2], where the definition of OWDV between any two packets a and b ($a \neq b$) is the difference of their OWDs, or

$$OWDV(a, b) = OWD(a) - OWD(b). \quad (1)$$

The problem is that the source and the destination generally use different clocks. Therefore, a major effort has been made in recent years to provide clock synchronization between remote hosts.

To describe clock synchronization, the following definitions are used in this paper. Clock A, C_A, has time t_A at the same time that Clock B, C_B, indicates t_B, where t_A and t_B are the indicated times or time stamps at the same time. The offset time is $O_{A,B} = t_A - t_B$. Clocks A and B advance at rate $r_A = d(t_A)/dt$ and $r_B = d(t_B)/dt$, respectively. Clock drift is the rate of change in the clock rate. For simplicity, this clock drift is assumed to be zero. Using C_A as the reference, clock skew is defined as $S_A = r_B/r_A$. In a similar way, the clock skew referenced at C_B is $S_B = r_A/r_B$.

Clock synchronization is the adjustment of time on the local clock relative to the reference's clock which may be at the source, destination, or some other node. In other words, clock synchronization is the elimination of the clock's offset.

There are two measurement approaches: active and passive. Active measurements inject probing packets into the network, specifically between source and destination hosts, with the purpose of measuring the network parameter of interest [3]-[5]. Several existing active measurement schemes use the global-positioning system (GPS), network time protocol (NTP) [6], or precision time protocol (PTP), also known as IEEE 1588 standard [7], to perform clock synchronization. However, the introduction of GPS increases equipment costs. Furthermore, GPS is difficult to deploy on legacy equipment. NTP and PTP require measurement of round-trip time (RTT) to perform clock synchronization, and even more restrictive, they require that both paths (from source to destination and from destination to source) have the same OWD to function. In general, each path has a different OWD unless the network was engineered with this feature in mind, and it is difficult to expect a network such as the Internet would be able to meet this restriction.

Active schemes that can measure OWD/OWDV without requiring clock synchronization have been proposed [8]-[10]. In [8], cyclic-paths are selected properly to calculate each one-way delay, while a new relationship is introduced between OWD and RTT in [9]. However, both are based on the measurement of round-trip delays. The scheme in [10] measures the time-interval values between pairs of packets at the sender and the receiver using dedicated UDP packets and calculates the difference between both values. These values are used to deduce the Fourier-transformed magnitude of the queuing delay probability distribution function and so avoid the clock synchronization problem. This scheme is very attractive. Unfortunately, it cannot realize packet-by-packet OWDV measurement.

Passive measurement schemes use the traffic flowing through the network, most often user traffic, to implement the measurement process. Passive measurement is attractive because it does not inject additional traffic into the network and because measurement is directly performed on the user traffic (the effect that the user actually experiences). However, the accuracy of this approach depends on how to capture the flowing traffic data at a given time. A passive measurement scheme to measure OWD was recently proposed [11]. This scheme uses GPS and NTP for clock synchronization. Another passive OWD scheme is based on the assumption that PTP achieves clock synchronization [12].

A scheme to measure OWD clock skew was recently proposed [13]. This scheme uses the real-time protocol (RTP) [14] and the real-time control protocol (RTCP) [15] to create fixed packet length and fixed inter-packet delay (IPD). IPD is the time between two consecutive packet departures at the source and is measured at the destination. This approach is based on a simple model to estimate clock skew, and demonstrates a different perspective from the existing complex approaches to the estimation of clock skew, where for example, linear programming is used [16]. The accuracy of clock skew removal depends on experiencing low queuing delays during the measurement. However, the case where the network remains with long queuing delays is not considered in the scheme. From the perspective of measurement accuracy, only GPS-based schemes are able to lower the error rate of clock frequency determination to under 10^{-8} [11]. Almost none of the methods based on clock skew estimation addressed the issue of measurement accuracy [8], [12], [16].

From the user's point of view, it must be possible to measure the OWD/OWDV of the user's data anywhere with no limitation. For example, it means that the tools are offering passive measurement and interoperability with general purpose data capturing tools such as Wireshark [17], and without the needs for clock synchronization. OWDV serves a very important role for jitter buffer design and analysis of network conditions [16]. This is because OWDV involves only queuing delays which strongly depend on network traffic conditions.

This paper proposes a passive OWDV measurement scheme that does not need an external mechanism for clock synchronization between the source and destination hosts. The proposed approach measures IPD and the combined and accumulated OWDV (AOWDV), which is the sum of IPD. However, as clock skew is included in the IPD measurements, this paper introduces an AOWDV-based mechanism to estimate and remove clock skew. OWDV is obtained from AOWDV after clock skew is removed. In this process, the proposed method estimates the clock-rate difference between the source and the destination. The proposed approach is evaluated by the results of both simulations and analyses of real network data. The evaluation results show that the proposed method can effectively estimate OWDV.

The remainder of this paper is organized as follows. Section II introduces the mechanism proposed to measure OWDV. This section also describes the methodology used for the detection and adjustment of clock skew. Section III presents a simulation study of the proposed scheme. Section IV presents the results of OWDV evaluations of data from an actual VoIP network. Section V presents our conclusions.

II. Scheme to Measure OWDV

1. Measuring OWDV

The basic idea behind the measurement of one-way delay is described below. For simplicity, we assume that there is no

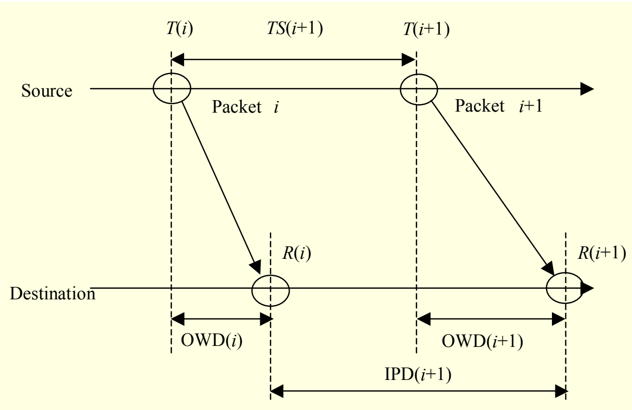

Fig. 1. Variables to measure OWDV based on IPD.

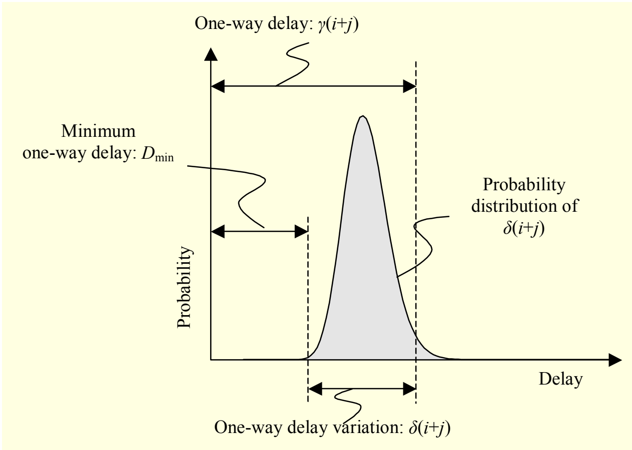

Fig. 2. Relationship of $\gamma(i+j)$, $\delta(i+j)$, and D_{min}.

clock skew in the network being examined. The following subsection focuses on clock skew removal.

The proposed mechanism is based on the differences between the IPD and TS measured at the destination host. The adopted method uses the smallest experienced difference to evaluate the contribution of every new IPD and TS measurement as packets are received at the destination. To evaluate the clock skew, the source explicitly sends packets together with departure times, where the departure time is measured at the source host, to the destination. The packets have fixed length to comply with the OWDV definition [2]. Every packet has an identification number. If i is defined as the first packet transmitted from the source to perform measurement, then $IPD(i+1)$ is measured between packets i and $i+1$. Figure 1 shows an example of the delay times between packets i and $i+1$ in a stream of packets. In this example, packet i departs at time $T(i)$, and packet $i+1$ departs at time $T(i+1)$, where $T(i)$ and $T(i+1)$ are measured at the source host. The arrival times of packets i and $i+1$ are then $R(i)$ and $R(i+1)$, as measured at the destination host.

The inter-packet delay at the source, $TS(i+1)$, is defined as

$$TS(i+1) = T(i+1) - T(i). \tag{2}$$

This value is set by the source. The inter-packet delay measured at the destination, $IPD(i+1)$, is

$$IPD(i+1) = R(i+1) - R(i). \tag{3}$$

This value is $TS(i+1)$ plus the variable $OWD(i+1)$ that a packet undergoes, and it is measured by reference to the clock at the destination.

In general, OWD consists of two types of delays: deterministic and variable delays. Propagation and transmission delays are deterministic. The propagation delay is defined as the propagation time over the transmission medium (optical or electrical signal) and is proportional to transmission length. The transmission delay is defined as the transmission time of a packet over the transmission link. It is calculated by dividing the packet length by the transmission link rate. These delays mentioned above do not vary unless source/destination hosts or communication paths in the networks and packet length are changed. Deterministic delays are treated as fixed delays, assuming that packet length is fixed.

Queuing delay is variable, and represents the times that packets wait at the buffers in intermediate network nodes for onward transmission. These delays depend on traffic conditions in the networks and show values equal to or greater than zero. If the queuing delay is equal to zero, the packet is not affected by any other traffic so that OWD takes its minimum value. The minimum OWD, which includes only fixed delays, is a critical factor in implementing the proposed scheme.

The measurement of OWDV is based on the measurement of IPD and TS. As shown in Fig. 1, $IPD(i+1)$ is measured as the difference in time between packets i and $i+1$. The minimum $OWD(i+j)$ experienced among all those measured, including that of packet $i+1$, is denoted as $D_{min.}$ The consideration of consecutive packets is not required as long as TS between two packets is conveyed to the destination host (there are several options for implementing this requirement [12]; however, they are outside the scope of this paper).

Figure 2 describes the relationships among OWD, OWDV, and D_{min}. Here, let the variables of OWD and OWDV be defined as $\gamma(k)$ and $\delta(k)$, respectively, for packet k hereafter. $\delta(i+j)$ experienced by packet $i+j$ in reference to D_{min} can be re-written as

$$\delta(i+j) = \gamma(i+j) - D_{min}. \tag{4}$$

On the other hand, the inter-packet delay at the source can be described as a function of $IPD(i+j)$:

$$TS(i+j) + \gamma(i+j) = \gamma(i+j-1) + IPD(i+j). \tag{5}$$

Using (5) and long measured periods, $\gamma(i+n)$ as OWD for the last received packet n can be expressed as

$$\gamma(i+n)=\sum_{j=1}^{n}[IPD(i+j)-TS(i+j)]+\gamma(i)\,, \quad (6)$$

where n is the index of the last measured OWD. Also, let the function AOWDV of variables of IPD, TS, and n, or $\sum_{j=1}^{n}(IPD(i+j)-TS(i+j))$ be defined as $\Delta(i+n)$. Therefore, (6) can be transformed into

$$\gamma(i+n)=\Delta(i+n)+\gamma(i). \quad (7)$$

$D_{\min}$ in (4) can be expressed from its definition and (7) as

$$D_{\min}=\min_{1\le j\le n}\{\gamma(i+j)\}=\min_{1\le j\le n}\{\Delta(i+j)\}+\gamma(i), \quad (8)$$

where $\min_{1\le j\le n}\{x_j\}$ is defined as a function of selecting the minimum value from the set of $\{x_1,x_2,\ldots x_n\}$. Using (4), (7), and (8), $\delta(i+n)$ for packet n can be expressed as

$$\delta(i+n)=\Delta(i+n)-\min_{1\le j\le n}\{\Delta(i+j)\}. \quad (9)$$

Equation (9) shows that the one-way delay of the first experience $\gamma(i)$ is removed. This means that the proposed scheme is able to estimate $\delta(i+n)$ by using measurable data of IPD and TS. However, since in practice these equations include clock skew information, the clock skew information has to be removed using a process described in the next subsection.

2. Removal of Clock Skew

The removal (also called adjustment) of the clock skew from $\Delta(i+j)$ measurements is described here. The adjustment needs to be done with reference to the destination clock C_d. Measurements of $TS(i+j)$ are made with the source clock, so they include information about the clock rate of C_s. Since the information does not include the difference of the departure times, clock synchronization is not needed for the clock rate of C_d. For this, $\Delta(i+j)$ is considered on a per-packet basis as in (9). Figure 3 shows a general description of how $\Delta(i+j)$ is collected. $\Delta(i+j)$ is calculated using $TS(i+j)$, $IPD(i+j)$, and $\Delta(i+j)$ when packet $i+j$ is received at the destination host. When this parameter is plotted against time (C_d), $\Delta(i+j)$ contains clock skew information that can be observed. However, $\Delta(i+j)$ values can include queuing delays, which render clock skew estimation inaccurate. A process that removes the queuing delays from the collected data of $\Delta(i+j)$ is required. For this, only the smallest values of $\Delta(i+j)$ selected in observation time units (defined as windows), $\min\{\Delta(i+j)\}$, including negative values, are used. The selection of $\min\{\Delta(i+j)\}$ is independent of the difference of the smallest value in the complete set of collected measurements (that is, n), which makes it dependent on user traffic. Instead, information on the rate of small values collected is used to divide the measurement time into windows whose size is determined by the rate of small values $r(l)$ collected in n samples under normalized traffic load (l), where the link or network capacity is used as reference value. That is,

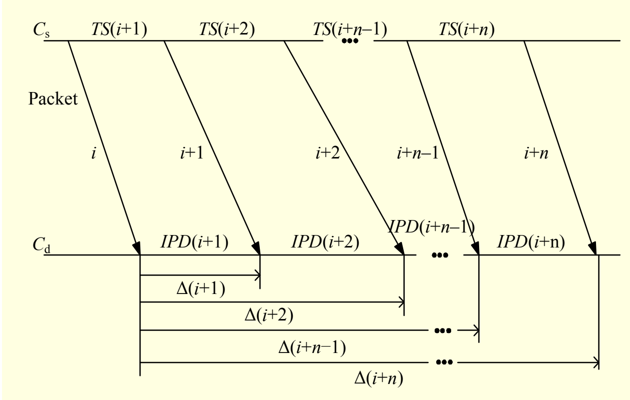

Fig. 3. Example of data collected for clock synchronization.

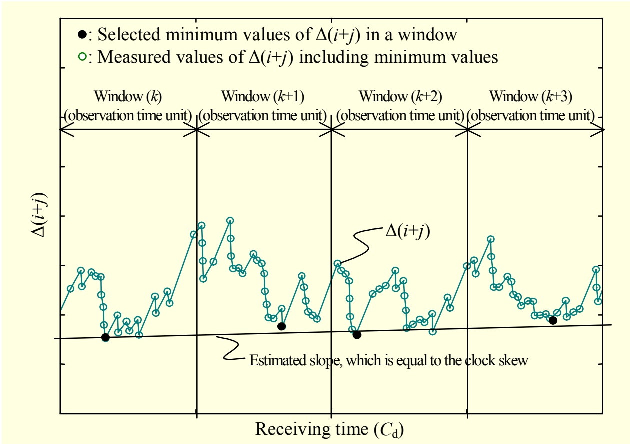

Fig. 4. Estimation process of clock skew.

$$r(l)=Ns(l)/n, \quad (10)$$

where $Ns(l)$ is the number of $\Delta(i+j)$ without queuing delay collected in n packets at load l. The window size is given in terms of the number of packets. There exists a trade-off between $r(l)$ and n. For this reason, the ns2 simulation [18] of the M/M/1 model, with l=0.8 (heavier traffic load condition), used the values of $r(l)$=0.01 and n=100. n=100 means a window size of two seconds. The rate of clock skew is estimated from the set of smallest values selected in every window by using linear regression. Figure 4 shows the estimation process. First, the minimum value in each window (k) is selected. Then, the slope, which is equal to the clock skew, is estimated using these minimum values based on linear regression. Finally, actual $\Delta(i+j)$ is obtained by subtracting the delay gaps due to clock skew from $\Delta(i+j)$.

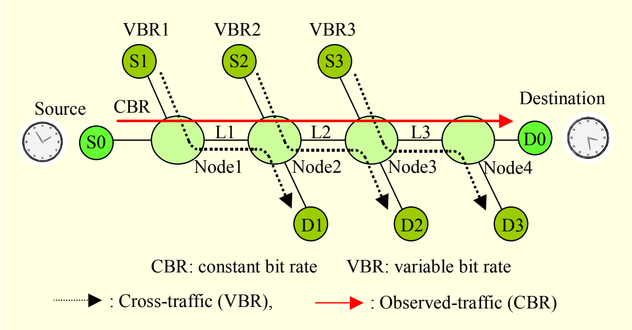

Fig. 5. Network simulation model for $\delta(i+j)$ estimation.

III. Simulation of Proposed Scheme

Network simulator ns2 was used to validate the proposed measurement scheme. We focused on validating the estimation of OWDV in IP networks with heavy cross traffic. Figure 5 shows the network model used in the simulations. The network consists of four source hosts (S0 to S3), four destination hosts (D0 to D3), four network nodes (Node1 to Node4), and three links (L1 to L3) connected between network nodes. All the links have the rate of 1 Mbps.

In this simulation model, a VoIP connection was selected to validate the OWDV measurement scheme in IP networks since VoIP calls have strict requirements in terms of OWDV. The VoIP traffic, which is transmitted from S0 to D0, is characterized by constant bit rate (CBR) with a 240-byte packet being transmitted every 20 ms. The VoIP traffic bandwidth can reach 96 kbps.

In addition, cross traffic flows were injected into the IP network. The flows are characterized as variable bit rate (VBR), random packet-length (average 700 bytes), and random packet-interval time to simulate aggregated TCP/IP traffic flows in the IP networks. Three independent cross traffic flows, VBR1, VBR2, and VBR3, were injected from source nodes S1, S2, and S3 and transmitted to destination nodes D1, D2, and D3, respectively. These flows impose independent and randomized cross traffic on the VoIP stream. The network loads, including the VoIP traffic over the links, were set to around 80%. The clock rate difference between the source and the destination was set to ±1000 parts per million (ppm), or ± 1 ms per s. This value is a stricter test than is expected in practice (order of 200 ppm at most).

Figure 6 plots the $\Delta(i+j)$ data calculated from the collected $TS(i+j)$ and $IPD(i+j)$ values, against the arrival times as clocked by C_d. In addition to the variability in the $\Delta(i+j)$ data caused by the queuing delays of the cross traffic, the figure shows that there is a steady increase in $\Delta(i+j)$. This offset increase is created by the clock skew.

To remove the clock skew, the smallest values of $\Delta(i+j)$ are

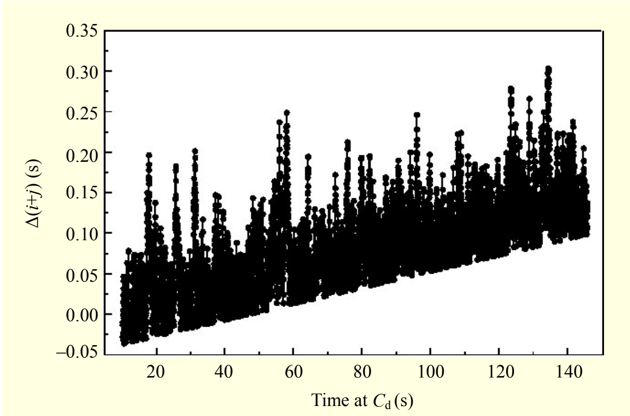

Fig. 6. $\Delta(i+j)$ with offset produced by clock skew between C_s and C_d.

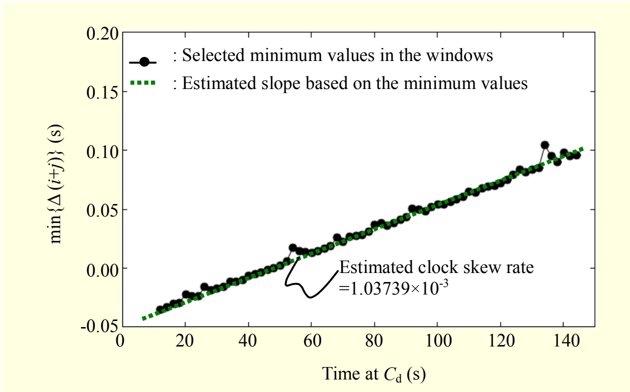

Fig. 7. Minimum $\Delta(i+j)$ values used for clock skew estimation.

collected, as defined in subsection II.2. Different from [13], where acceptable error ε from the minimum experienced delay is used to determine the selection of delays used in calculating the clock skew, the approach adopted in this paper segments the simulation time into windows, and the lowest value is collected from each window. The windows selected here have a length of 100 packets, or 2 s (as CBR traffic is used).

Figure 7 shows the minimum values of $\Delta(i+j)$ collected in the case in Fig. 6 ($C_d > C_s$). The estimated skew rate is represented by the slope of the straight line in the figure. These values show similar increasing linear offset to that observed in Fig. 6. The advantage of this method is that even if queuing delay remains constant, the collectable $\Delta(i+j)$ values can be used for the calculation of the clock skew.

The clock skew, estimated by using linear regression, was 1.03739×10^{-3}. This means that the calculated clock skew has an error of 3.7% (against the 1.000×10^{-3} of the experimental setup). After the skew is calculated, it is removed from the measured $\delta(i+j)$. This is the actual $\delta(i+j)$.

Figure 8 shows actual $\delta(i+j)$ in this experiment, where the offset created by the clock skew is removed. As the figure

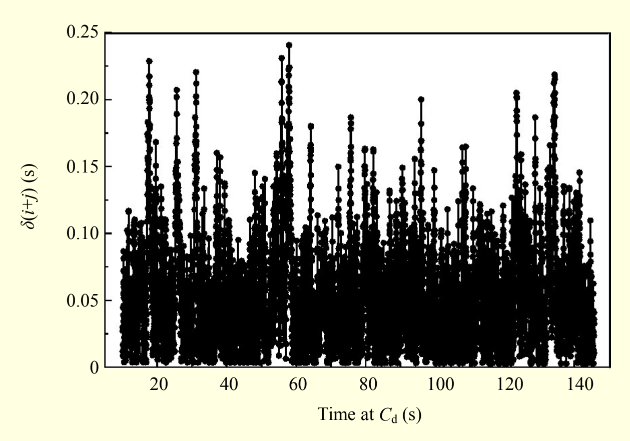

Fig. 8. $\delta(i+j)$ without clock skew.

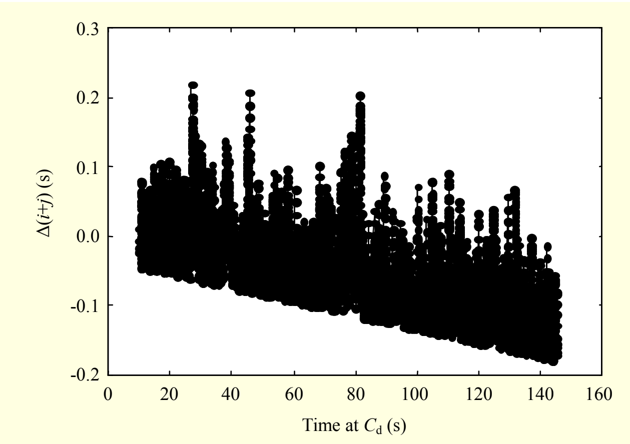

Fig. 9. $\Delta(i+j)$ with offset produced by clock skew between C_s and C_d.

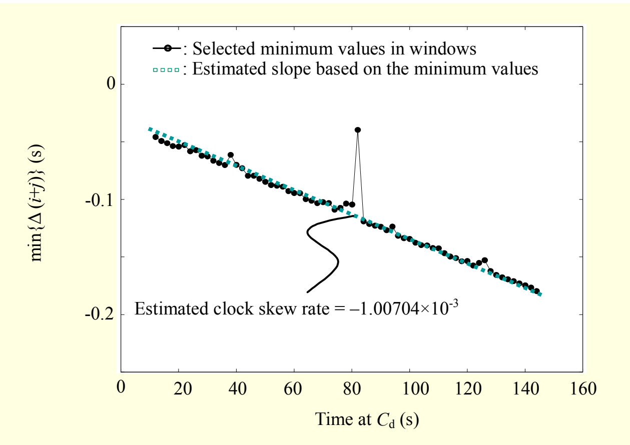

Fig. 10. Minimum $\Delta(i+j)$ values used for clock skew estimation

shows, the distribution is similar to that shown in Fig. 6, so no information is lost.

A simulation of the case of $C_d < C_s$ was also conducted. Figure 9 shows the $\Delta(i+j)$ data at the same link utilization rate (80%) as in the case of Fig. 6; a negative clock skew rate (–1000 ppm) and longer packet length (average 1000 bytes) were used. The longer packet length raises the probability of greater queuing delay.

We can see that there are larger $\Delta(i+j)$ values (more than 300 ms) in Fig. 9 than in Fig. 6. Figure 10 shows the minimum values of $\Delta(i+j)$ based on the data shown in Fig. 9. The clock skew rate estimated using the same method is represented by the slope of the straight line in the figure. These values yield a decreasing linear offset. As shown in Fig. 10, a large value, which included long queuing delay, was detected at around 80 seconds. This was caused by high traffic loads with longer packet length. Almost all values seem to be close to the actual minimum $\Delta(i+j)$, so the slope of clock skew could be obtained with high accuracy. Estimations for $\delta(i+j)$ values were done in the same way as for the case of $C_d > C_s$. The figures for $\delta(i+j)$ are omitted due to space concerns.

IV. Field Test of Proposed Scheme

OWD measurements were conducted in a practical field environment. The system used offers private VoIP services to around 20,000 IP phones in major Japanese cities. The distance between the offices with IP phones and the network operation center (NOC) is as long as 2,000 km. The networks consist of various network subsystems, such as private core IP networks, public VoIP networks, Internet, ADSL, and Wireless LAN (IEEE 802.11). The network systems, including IP phones, are run without clock synchronization.

VoIP quality metrics such as packet loss and jitter (except one-way delay (variation)) were monitored at the NOC. Figure 11 shows the system used to measure $\delta(i+j)$ of the transmitted voice packets. The voice packets on the monitored paths were copied and transferred to packet capture tools installed at the NOC. The packet capture tools record the information of proper RTP packets, including transmission (timestamp) and reception times. Here the proper packets are defined as packets received with no protocol violation. Information of VoIP calls was stored in the packet capture tools to estimate $\delta(i+j)$. The source of this information is indicated by the red dotted lines in the figure.

The first example is that the observed VoIP call was impacted by the bursty cross traffic on the Internet during the calling time from 18 s to 105 s. We did not analyze the cause of the bursty traffic because it is outside our scope.

Figure 12 shows the estimated $\Delta(i+j)$ based on data sampled at the NOC. Indicating that clock skew was small, $\Delta(i+j)$ shows an apparent constant offset. The collected minimum values of $\Delta(i+j)$, as mentioned in subsection II.2, are also shown in Fig. 13, where the estimated skew rate is represented

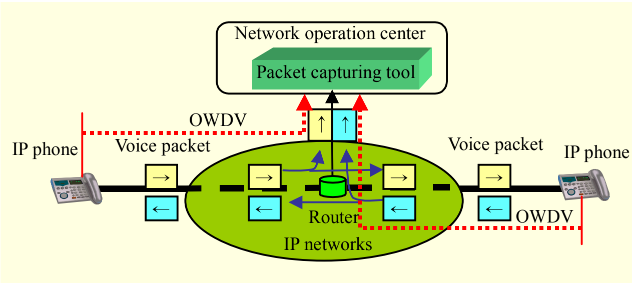

Fig. 11. Network model for field measurement of $\delta(i+j)$.

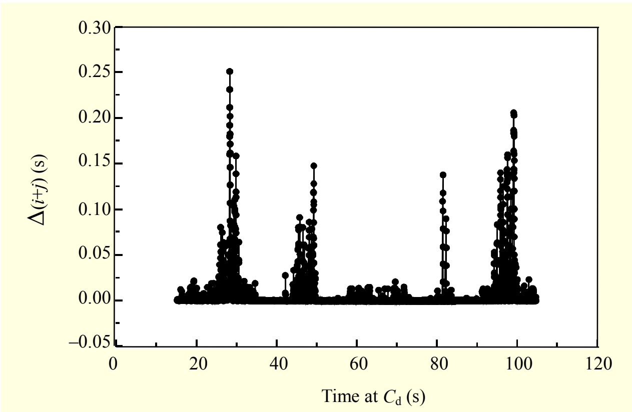

Fig. 12. $\Delta(i+j)$ values collected in actual VoIP network.

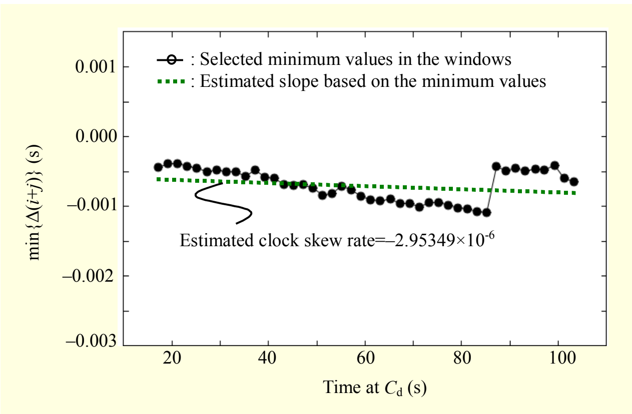

Fig. 13. Minimum $\Delta(i+j)$ values from first experiment on actual VoIP network.

by the slope of the straight line. The clock skew rate was estimated, by linear regression, to be -2.95349×10^{-6}. This shows that C_d has a slightly smaller rate than C_s due to having a minus rate. Since the absolute rate of 2.95349×10^{-6} corresponds to around 10 ppm, clock skew between the source and destination is small compared to the specification requirement (order of 100 ppm). Due to the duration of the collection process, the values show a variation that could indicate the magnitude of the clock skew. However, the largest values are the most representative.

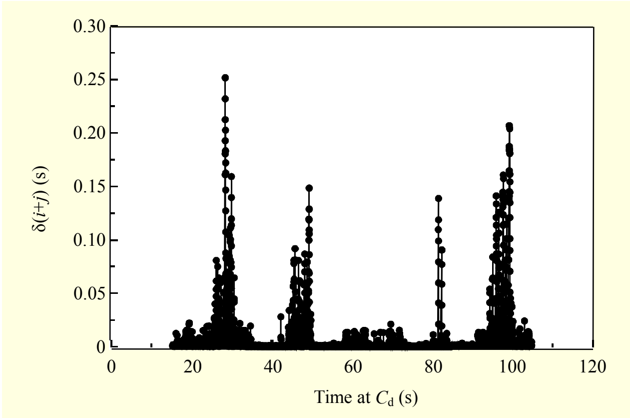

Fig. 14. $\delta(i+j)$ from the first experiment on actual VoIP network.

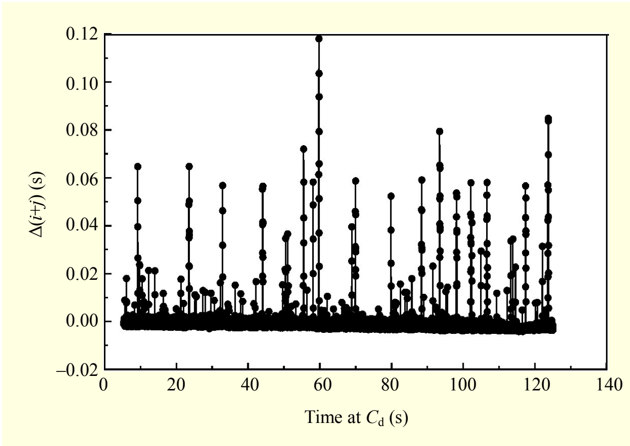

Fig. 15. $\Delta(i+j)$ values of actual VoIP networks; second experiment.

Figure 14 shows $\delta(i+j)$ values after clock skew removal based on the data of $\Delta(i+j)$ shown in Fig. 12. $\delta(i+j)$ in Fig. 14 consists of positive data and shows estimated $\Delta(i+j)$. In this case, the small clock skew looks to be easily removed by a simple process, and data from other trials show a similar trend. However, linear regression is used for this purpose, and it is applied on minimum values.

The second experiment yielded the case of packet loss (at around 35 seconds) while packets were randomly delayed in all connection periods in the wireless network (IEEE 802.11), which is located at the office. This data shows a smaller variation of the offset values indicated the clock skew. Figure 15 shows the $\Delta(i+j)$ values gathered in this experiment. The offset values have a small slope, indicating that this path between the IP phone and the capture tool had relatively small clock skew.

The minimum values of $\Delta(i+j)$, used to calculate the clock skew, are shown in Fig. 16. The estimated clock skew rate is represented by the slope of the straight line. The estimated clock skew rate shows that the absolute rate (16.9733×10^{-6}) is larger than that in Fig. 13, and C_d has a slightly smaller rate

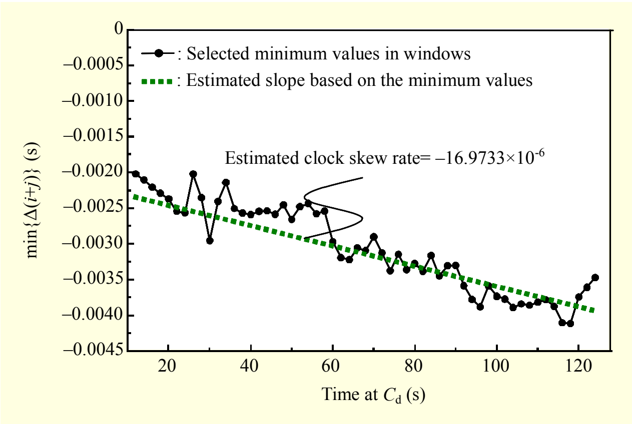

Fig. 16. Minimum Δ(*i*+*j*) values of actual VoIP network; second experiment.

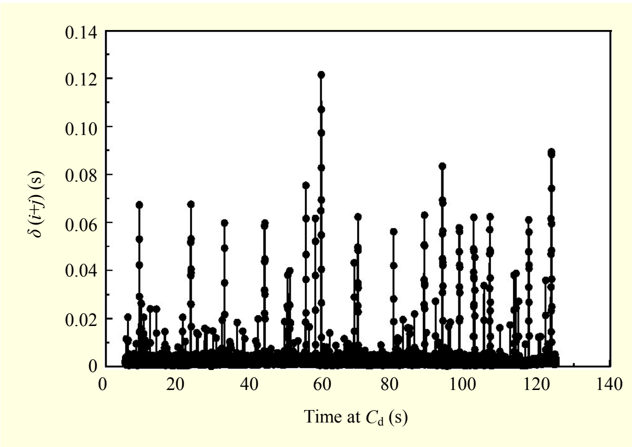

Fig. 17. *δ*(*i*+*j*) of actual VoIP network; second experiment.

than C_s due to having a minus rate the same as in Fig. 13.

Figure 17 shows *δ*(*i*+*j*) values after clock skew removal. The figure shows Δ(*i*+*j*) characteristics that match those shown in Fig. 15. As shown, the clock skew is effectively removed without degrading network information. This result shows that the proposed scheme performed well even if some packets were lost. This is due to the fact that Δ(*i*+*j*) can always be calculated by using pairs of *TS*(*i*+*j*) and *IPD*(*i*+*j*) values .

The proposed scheme was used to analyze OWDV of VoIP calls including calls with extremely poor voice quality (longer delay and/or many packet losses) in addition to the two examples mentioned above. The results confirm that OWDV measurement is useful for the analysis of network conditions like congestion.

V. Conclusion

OWDV, which is experienced by packets traveling between source and destination hosts, can be easily measured if clock synchronization is performed between the two hosts. Clock synchronization is known to be difficult except for the GPS-based system because time adjustment protocols require symmetrical paths for round trip measurements. However, as legacy equipment may need to be completely replaced to deploy GPS, this approach may be too slow or expensive. As an alternative that uses no additional resources, this paper presents a scheme for measuring OWDV that does not require an external mechanism for clock synchronization. The method depends on clock skew adjustment. This paper presents the dependability of estimating OWDV from the clock skew. The proposed scheme is based on the measurement of IPDs, which are the difference in arrival times of consecutive packets. The IPD values are closely related to OWDV. However, they are affected by the clock skew between the source and destination clocks. This paper introduced a method for the estimation and detection of clock skew to enable the measurement of OWDV. Different from existing methods, our method to estimate the clock skew is based on the accumulated values of OWDV or AOWDV. The use of AOWDV speeds up the collection of samples as it does not discard useful data as is done by error-based methods.

The proposed measurement scheme was tested by computer simulation and in an actual VoIP network. The results of the simulations and field trials show that clock skew can be effectively removed, and the OWDV values can be efficiently obtained. The error in clock skew estimation achieved was equal to or smaller than 3.7%. The proposed scheme can be applied by network operators to detect large jitter values or OWDV and to make network adjustments.

References

[1] J. Wang, M. Zhou, and Y. Li "Survey on the End-to-End Internet Delay Measurements," *High Speed Networks Multimedia Communications*, Springer, 2004.

[2] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)," RFC 3393, Nov. 2002.

[3] Y. Shavitt et al., "Large Scale Internet Queueing Delay Tomography," *Proc. IEEE INFOCOM*, Barcelona, Catalunya, SPAIN, Apr. 23-29, 2006.

[4] T. Iwama et al., "Real-Time Measurement of One-Way Delay in the Internet Environment," *Proc. Commun. Soc. Conf. IEICE*, B-16-1, Sept. 2004.

[5] U. Hofmann et al., "One-Way-Delay Measurements with CM Toolset," *Proc. IEEE Int. Conf. Performance, Computing, Commun.*, Phoenix, Arizona, Feb. 20-22, 2000.

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

[7] C. Gordon, "Introduction to IEEE 1588 and Transparent Clocks,"

White Paper, Tekron, 2009.
[8] O. Gurewitz and M. Sidi, "Estimating One-Way Delays From Cyclic-Path Delay Measurements," *INFOCOM*, Anchorage, Alaska, USA, Apr. 22-26, 2001.
[9] D. Kim and J. Lee, "End-to-End One-Way Delay Estimation Using One-Way Delay Variation and Round-Trip Time," *Proc. Qshine*, 2007.
[10] W.Z. Lu, W.X. Gu, and S.Z. Yu, "One-Way Queuing Delay Measurement and Its Application on Detecting DDoS Attack," *J. Netw. Computer Appl.*, vol. 32, no. 2, Mar. 2009, pp. 367-376.
[11] T. Zseby et al., "Passive One-Way-Delay Measurement and Data Export," *Proc. Int. Workshop Inter-domain Performance Simulation*," Salzburg, Austria, Feb. 20-21, 2003.
[12] M. Cola et al., "Covert Channel for One-Way Delay Measurements," *Proc. Int. Conf. Computer Commun. Netw.*, San Francisco, CA, USA, Aug. 3-6, 2009.
[13] B. Ngamwongwattana and R. Thompson, "Measuring One-Way Delay of VoIP Packets without Clock Synchronization," *Proc. IEEE Int. Instrumen. Meas. Technol. Conf.*, Singapore, May 5-7, 2009.
[14] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, Jul. 2003.
[15] C. Huitema, "Real Time Control Protocol (RTCP) Attribute in Session Description Protocol (SDP)," RFC 3605, Oct. 2003.
[16] S.B. Moon, P. Skelly, and D. Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements," *IEEE INFOCOM*, vol. 1, NY, USA, Mar. 21-25, 1999.
[17] Wireshark Network Protocol Analyzer. Available: http://www.wireshark.org/
[18] ns2 Network Simulator. Available: http://www.isi.edu/nsnam/ns, Jul. 2009.

Makoto Aoki received the BS in electrical engineering from Tokyo University of Science, Tokyo, Japan, in 1971. Since joining NTT in 1971, he has worked in the Switching System Department doing research and development of remote control switching systems, data communication systems, communications protocol control LSI (X.25 and high-speed packet communications protocol) and so on. From 1998 to 2007, he worked with Infonex and NTT-data Customer Service, where he was responsible for the multimedia packet network/system design for the company's networks. He joined Cyber Creative Institute in 2007 and mainly conducts research/consultation about VoIP networks, next generation network, IPTV and so on. His research interests are quality-of-service networks, network measurements, and high-speed communications protocol control for future Internet. He is an IEEE member and IEICE member.

Eiji Oki is an associate professor of The University of Electro-Communications, Tokyo, Japan. He received his BE and ME in instrumentation engineering and the PhD in electrical engineering from Keio University, Yokohama, Japan, in 1991, 1993, and 1999, respectively. In 1993, he joined NTT's Communication Switching Laboratories, Tokyo, Japan. He has researched IP and optical network architectures, traffic-control methods, high-speed switching systems, and communications protocols. From 2000 to 2001, he was a visiting scholar at Polytechnic University, Brooklyn, New York, where he was involved in designing tera-bit switch/router systems. He joined the University of Electro-Communications, Tokyo, Japan, in July 2008. Dr. Oki was the recipient of the 1998 Switching System Research Award, the 1999 Excellent Paper Award presented by IEICE, and the 2001 Asia-Pacific Outstanding Young Researcher Award presented by the IEEE Communications Society for his contribution to broadband network, ATM, and optical IP technologies. He has co-authored two books, *Broadband Packet Switching Technologies*, published by John Wiley, New York, in 2001 and *GMPLS Technologies*, published by RC Press, Boca Raton, in 2005. He is an IEEE senior member and an IEICE senior member.

Roberto Rojas-Cessa received the MSc and PhD in computer engineering and electrical engineering from Polytechnic Institute of New York University, Brooklyn, NY, respectively. Currently, he is an associate professor at the Department of Electrical and Computer Engineering, in New Jersey Institute of Technology, Newark, NJ. He has been involved in the design and implementation of application-specific integrated-circuits for applications in biomedical equipment and high-speed computer communications, and in the development of high-performance and scalable packet switches and switch reliability. He worked on the development of a commercial terabit-range core packet switch and in the development of FPGA designs for high-computation high-throughput applications in Core Networks, Inc., Tinton Falls, NJ. He was a research associate and an adjunct professor in Polytechnic Institute of NYU where he worked on the design of multi-terabit packet switches. His research interests are high-speed switching and routing, reliability and fault tolerance, quality-of-service networks, network measurements, and high-layer protocols. His research has been supported by the USA National Science Foundation and private companies. He enjoys serving on IEEE technical committees, organizing IEEE conferences, and as a panelist for the US National Science Foundation and the US Department of Energy.