A Relative Time Model in a Distributed Network using Exchanged Time Information

Cheng Guang, Gong Jian, Ding Wei
(Computer Department of Southeast University, Nanjing 210096, P.R.China)
(gcheng@njnet.edu.cn)

Abstract: In a distributed network, a great number of network applications can be performed by means of measuring accurate synchronization clock from different network computers, such as measurements of one-way delay. Due to the difference of frequency and time between different clocks, these clocks are non-synchronization, so one-way delay measurements between two computers will exist exists relative time offset. [7, 8, 9] directly analyze one-way delay measurements to remove measurement error. But in the paper, the information of frequency and time between computers clocks is exchanged, and the relative clock model between two computer clocks can be established, based on the relative clock model, the measured delay timestamp is corrected, and so the one-way delay measurements is also corrected indirectly. Two groups of measured data in a local area network, and one group of one-way delay between Southeast University in China and AMS in Switzerland are analyzed to verify the relative clock model. The result shows that the relative clock model can describe the relative relationship between different computer clocks, and correct the measured error of one-way delay between computers.

Keywords: Synchronization, Relative Clock Model, One-way Delay, Relative Skew, Network Performance

1 Introduction

Precise clock synchronization is frequently used to measure one-way delay. The clocks on the network hosts used to measure the times, however, are not always synchronized, and this lack of synchronization reduces the accuracy of these measurements. Therefore, estimating and removing relative frequency skews and time offsets model from exchanging time information between sender and receiver clocks are critical to the accurate assessment and analysis of one-way delay.

In the paper we introduce the NTP protocol technology to exchange time information between sender and receiver clocks, and to hypothesis that one-way delay from sender to receiver is the same as that from sender to receiver. NTP protocol analysis the clock relation between client and server to build the clock model in the local client, however the paper considers the equal relation between two computer clocks to build the relative relation model between two clocks.

The relative relation clock model is used to measure the one-way delay [3]. Measuring one-way delay has two kind of ways both active measuring method [4, 5] and passive measuring method [6], However, keeping the relative clock synchronization between sender and receiver is critical to measure one-way delay using everyone measuring method. Now the two synchronization technology both NTP and GPS is to correct the absolute clock of client, so measurements of one-way delay exists relative clock offset. In order to estimate and remove relative skews and offsets, [7, 8, 9] introduce different algorithms to estimate the clock skew in network one-way delay measurements. But these algorithms correct one-way delay measurements directly, so these algorithms have high time-complexity, and cannot remove relative clock offsets and skews entirely. The difference of the paper is to research the clock relation of two end computers of one-way delay measurements. According to the clock relation, a relative time model is modeled, and based on the relative time model, the one-way delay measurements can be corrected.

The rest of the paper is organized as follows. In Section II we analysis the relative skews and offsets between computer clocks to establish the relative clock skew-offset model. In Section III we measure three groups of exchanging information data between sender and receiver. Two groups of data are measured inside the njnet.edu.cn network, and one group data comes from the AMS project [10] that the data is measured between njnet.edu.cn in China and cern.ch in Switzerland. The three groups of data are used to verify the model performance, as well as its estimating error. In Section IV, we rely on the relative time model to establish a correcting one-way delay model, and the AMS program data is used to experience the correcting one-way delay model. We conclude the paper in Section V.

0-7803-8145-9/04/$17.00 ©2004 IEEE.

2 Relative Time Concepts and Relation

Suppose that a "true" clock is accurate at any moment, and runs at a constant rate. Let the "true" clock $C_t(t) = t \geq 0$, and C1 and C2 be two clocks:

- **Relative Offset:** C1(t) – C2(t), the difference between the time reported by C1 and C2
- **Relative Skew:** the difference in the frequencies of C1 and C2, denoted as $(C1'(t) - C2'(t))$.
- **Relative Drift:** the drift of C1 is $C1''(t)$, and the drift of C1 relative to C2 at time t is $(C1''(t) - C2''(t))$.
- **Relative Clock Synchronization:** Both C1 and C2 are synchronized at a particular moment t_0 if both the relative offset and skew are zero, or less than an appointed threshold.
- **Relative Time:** Let C1 = t1, and C2 = t2 at a "true" time t_0, so we call that t1 of C1 relative to time of C2 is t2.

Let f(t) be the time displayed by a clock at epoch t relative to the standard timescale:

$$C(t) = \alpha(t - t_0)^2 + \beta(t - t_0) + t_c + x(t) \quad (1)$$

Where α is the fractional frequency drift per unit time, β is the frequency, t_c is the time at some previous epoch t_0, and the random nature of the clock is characterized by x(t) which represents the random noise relative to the "true" time. Every clock model can be characterized by equation (1), so the time of both clock C1 and C2 at the "true" time t can be modeled as equation (2) and (3) respectively.

$$C1(t) = \alpha_{C1}(t - t_0)^2 + \beta_{C1}(t - t_0) + t_c^{C1} + x_{C1}(t) \quad (2)$$

$$C2(t) = \alpha_{C2}(t - t_0)^2 + \beta_{C2}(t - t_0) + t_c^{C2} + x_{C2}(t) \quad (3)$$

And the relative offset between C1 and C2 is

$$\begin{aligned} C1(t) - C2(t) &= (\alpha_{C1} - \alpha_{C2})(t - t_0)^2 \\ &+ (\beta_{C1} - \beta_{C2})(t - t_0) + t_c^{C2} - t_c^{C1} + x_{C1C2}(t) \end{aligned} \quad (4)$$

Usually the second-order $\alpha_{C1} - \alpha_{C2}$ is ignored and the noise term $x_{C1C2}(t)$ is modeled as a normal distribution with predictable autocorrelation function. So the equation (4) can be simplified as equation (5).

$$C1(t) - C2(t) = (\beta_{C1} - \beta_{C2})(t - t_0) + t_c^{C1} - t_c^{C2} \quad (5)$$

And the equation (2) and (3) also are simplified as equation (6) and (7).

$$C1(t) = \beta_{C1}(t - t_0) + t_c^{C1} \quad (6)$$

$$C2(t) = \beta_{C2}(t - t_0) + t_c^{C2} \quad (7)$$

According to function (6) and (7), we can obtain the relative time of both C1 and C2.

$$C2(t) = \frac{\beta_{C2}}{\beta_{C1}} C1(t) + t_C^{C2} = \lambda C1(t) + t_C \quad (8)$$

And similarly we can have

$$C1(t) = \frac{1}{\lambda} C2(t) - \frac{1}{\lambda} t_C \quad (9)$$

where $\lambda = \frac{\beta_{C2}}{\beta_{C1}}, t_C = t_c^{C2} - \frac{\beta_{C2}}{\beta_{C1}} t_c^{C1},$. Equation (8) characterizes the relative time of C2 to C1, and equation (9) characterizes the time of C1 relative to C2. According to the equation (8) or (9), the relative skew model between C1 and C2can be as following:

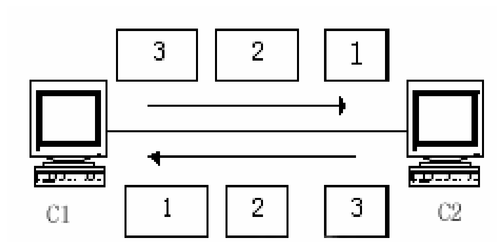
$$C2(t) - C1(t) = (\lambda - 1)C1(t) + t_C \quad (10)$$

3. Relative Clock Model

In this section, firstly we will research how to exchange time information between C1 computer and C2 computer. Secondly, a relative time model between C1 and C2 is analyzed and established by the time information.

Figure 1 shows the architecture to exchange the time information between C1 computer and C2 computer. In the architecture, C1 computer sends timestamp request packet with the C1 timestamp to C2 computer. As soon as C2 computer receives the packet, C2 computer sends the timestamp reply packet to C1 computer. In the paper, the exchanging time information tool is a modified version of

Timestamp Request Packet



Timestamp Reply Packet

Fig 1 Exchanging Time Information Model

Mtools [12] that is a collection of tools for measuring network performances, and is made up of two instruments: one-way-delay meter, and round-trip-time meter. After sending a UDP timestamp packet to the receiver site, the measuring host waits for the destination to reply a UDP timestamp packet, which involves four timestamp fields. Figure 2 shows the UDP timestamp packet structure.

0   63
IP Packet Head (20 byte)
UDP Packet Head (8 byte)
Originate Timestamp
Received Timestamp
Transited Timestamp
Finished Timestamp

Figure 2: UDP timestamp packet

$C1(t_{i,1})$: Before sending the UDP timestamp request packet, the source (computer C1) puts its current time value into the originate timestamp field of the UDP packet.

$C2(t_{i,2})$: After receiving the timestamp request packet, the destination (computer C2) inserts its current time value into received timestamp field of the UDP packet.

$C2(t_{i,3})$: Before sending the UDP timestamp reply packet, the destination puts its current time into the transited timestamp field.

$C1(t_{i,4})$: After received the timestamp reply packet, the source inserts its current time value into fourth timestamp field of the UDP packet.

Therefore, in the source, four timestamps $C1(t_{i,1})$, $C2(t_{i,2})$, $C2(t_{i,3})$, $C1(t_{i,4})$ can be obtained, and their accurate time is $t_{i,1}$, $t_{i,2}$, $t_{i,3}$, $t_{i,4}$ respectively. Where i means to measure i^{th} timestamp packet, $1 \leqslant i \leqslant n$, and n is the number of the measuring timestamp packet. D. Mills suggested that 8 time packets are exchanged in NTP [1], so we suggest that n is equal to 8 at least.

Symmetrical Delay Hypothesis : in figure 1, one-way delay of the UDP timestamp packet from C1 to C2 is as much as that from C2 to C1, that is

$$delay_{i,C1\_C2} = delay_{i,C2\_C1} \qquad (11)$$

Where $delay_{i,C1\_C2}$ is one-way-delay of i^{th} timestamp packet from C1 to C2, $delay_{i,C2\_C1}$ means one-way-delay of i^{th} timestamp packet from C2 to C1. The differences between timestamps can be used as the indicators of one-way delay, shown as following

$$delay_{i,C1\_C2} = t_{i,2} - t_{i,1} \qquad (12)$$

$$delay_{i,C2\_C1} = t_{i,4} - t_{i,3} \qquad (13)$$

According to Symmetrical Delay Hypothesis, so that

$$t_{i,4} - t_{i,3} = t_{i,2} - t_{i,1} \qquad (14)$$

Replacing the accurate time $t_{i,1}$, $t_{i,2}$, $t_{i,3}$, $t_{i,4}$ with the equation (6) and (7), we can have

$$\frac{(C1(t_{i,4})+C1(t_{i,1})-2t_c^{C1}}{\beta_{C1}} = \frac{(C2(t_{i,3})+C2(t_{i,2}))-2t_c^{C2}}{\beta_{C2}}$$

$$\frac{C1(t_{i,4})+C2(t_{i,1})}{2} = \frac{\beta_{C1}}{\beta_{C2}}\left(\frac{C2(t_{i,3})+C2(t_{i,2})}{2} - t_c^{C2}\right) + t_c^{C1} = \frac{1}{\lambda}\frac{C2(t_{i,3})+C2(t_{i,2})}{2} - \frac{1}{\lambda}t_C \qquad (15)$$

$$\frac{C2(t_{i,3})+C2(t_{i,2})}{2} = \lambda\frac{C1(t_{i.4})+C1(t_{i,1})}{2} + t_C \qquad (16)$$

The figure 3 shows the physical meaning of equation (15) and (16), that is, the median between sent packet time and finished packet time in computer C1 is equal to the median between received packet time and transit*ed* packet time in computer C2. If we measure n timestamp packets, a group of time pair $\left(\frac{C1(t_{i,4})+C1(t_{i,1})}{2}, \frac{C2(t_{i,3})+C2(t_{i,2})}{2}\right)$ can be obtained from the measured timestamp time series. The parameter λ and t_c of the equation (15) or (16) using least squares method, and

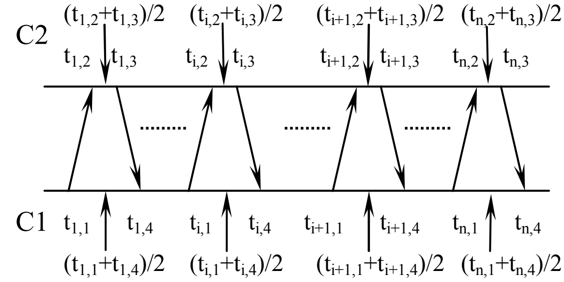


Figure 3 Timestamps Time Series

the relative time model can be expressed in the equation (17) and (18).

$$C2(t) = \lambda \times C1(t) + t_C \qquad (17)$$

$$C1(t) = \frac{1}{\lambda} C2(t) - \frac{1}{\lambda} t_C \qquad (18)$$

In the next section, two groups of experiment data will prove the linear relative time model can simulate the measured data very well, and the model estimated error don't influence the one-way delay measurements.

4 Experiment Analysis

We have measured two groups of data between sender and receiver to verify the performance as well as their estimating error of relative clock models defined by (15) and (16). Two groups of data (Traffic1 and Traffic2) were measured in the testbed of our lab.

In the experiment, the measured timestamps are the time of C1 and C2, if these timestamps is used directly, estimated model error will be increased. So we will define an originated time t_{C1} and t_{C2} of both C1 and C2, so the group of time pair is changed $\left(\frac{C1(t_{i,4}) + C1(t_{i,1})}{2} - t_{C1}, \frac{C2(t_{i,3}) + C2(t_{i,2})}{2} - t_{C2} \right)$. In the two groups of data measured in the testbed of our lab, both t_{C1}=1036811934 and t_{C2}= 1036811952 are the original time of C1 and C2 in the model respectively. In the figure 4, the X axis shows $\frac{C1(t_{i,4}) + C1(t_{i,1})}{2} - t_{C1}$, and the Y axis shows $\frac{C1(t_{i,4}) + C1(t_{i,1})}{2} - t_{C1}$.

Figure 4 shows the relative time relation of both C1 and C2 in Traffic1 data, with least squares method, where λ =1.000101, t_C=1.601516.

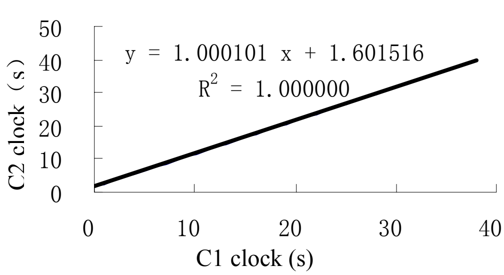


Fig 4 Traffic1 data

In order to estimate the performance of relative time model in measurements and simulations, R^2 metrics is defined as following:

$$R^2 = 1 - SSE / SST \qquad (19)$$

Where $SSE = \sum (y_i - \hat{y})^2$, $SST = \left(\sum y_i^2\right) - \left(\sum y_i\right)^2 / n$,

$R^2 \in [0, 1]$. If R^2 approaches to 1, the model will have a good simulation effect. In figure 4, R^2 = 1.000000 that shows the relative time linear model can almost simulate the measuring timestamp data fully, let $C1'(t) = C1(t) - 1036811934$, and $C2'(t) = C2(t) - 1036811952$, so that we can have

$$C1'(t) = 0.999899 C2'(t) - 1.601354 \qquad (20)$$

$$C2'(t) = 1.000101 C1'(t) + 1.601516 \qquad (21)$$

Equation (19) is the relative time of C1 to C2, and the equation (20) is the relative time of C2 to C1.

Figure 5 shows the relative relation of C1 and C2 in Traffic2 data, and equation (21) and (22) are simulated model of the Traffic2 data.

$$C1'(t) = 0.999900 C2'(t) - 1.602254 \qquad (22)$$

$$C2'(t) = 1.000100 C1'(t) + 1.602414 \qquad (23)$$

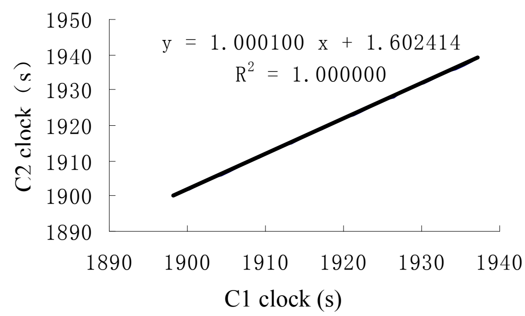


Fig 5 Traffic2 data

The interval between the Traffic1 data and the Traffic2 data is about 1900s. The relative time model has been changed, that including the relative offset and the relative skew. The relative skew is changed 0.000001, and the relative offset is changed 0.000898s. The reason why model parameters are

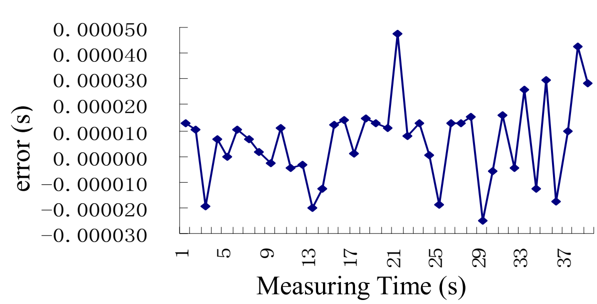


Fig 6: error time-series of model (20)

changed is the second-order team is ignored.

RFC1305 clock model considers the noise term as a normal distribution. The estimating error of the relative time model in the paper is composed of the linear model error and the noise term error. Figure 6 shows the error time-series (20), whose maximal error is 0.048ms, median error is equal to 0.006ms, and standard squared error is 0.0026ms, so the relative time model can be used to estimate and remove relative skews and offsets of the one-way delay.

5 Corrected One-way Delay

End-to-End one-way delay is frequently used to analyze network performance. The accuracy of measuring one-way delay is very important for many network protocols and applications based their control on observed network performance. For measurements of one-way delay, the sender needs to insert its timestamps into packets, so that the receiver can gather one-way delay information. Because the clocks at both end-systems are involved in measuring one-way delay, time synchronization of the two clocks becomes a focus in the accuracy of one-way delay measurements.

NTP protocol is widely used in the Internet for clock synchronization, and provides accuracy of the order of milliseconds under LAN circumstances. To obtain an accurate measurement of one-way delay, relative offset and skew between two clocks need to be accounted for. When two clocks involved in measurements of the one-way delay run at different frequencies and time, inaccuracies are introduced into the measurements. V. paxson focused on filtering out the effects of clock relative skew and offset specifically in one-way delay measurements. In the paper, we prove a corrected time model based the relative time model, to assure that inaccuracies of time and frequency are not introduced into measurements of one-way delay.

In the Section II, the source (computer C1) can obtain four timestamps $C1(t_{i,1}), C2(t_{i,2}), C2(t_{i,3}), C1(t_{i,4})$, where $C1(t_{i,1}), C1(t_{i,4})$ is time of clock C1, and $C2(t_{i,2}), C2(t_{i,3})$ is time of clock C2. Since the frequency and time of both clock C1 and clock C2 are different, one-way delay between computer C1 and computer C2 can not be measured directly by means of the four timestamps. We need to transfer timestamps of different clocks into timestamps of one same clock. Equation (24) and (25) can transfer timestamp of clock C1 into timestamp of clock C2 or vice verse.

$$C2(t_{i,2}'^{C1}) = \lambda C2(t_{i,2}) + t_c \, , C2(t_{i,3}'^{C1}) = \lambda C2(t_{i,3}) + t_c \qquad (24)$$

$$C1(t_{i,1}'^{C2}) = \frac{1}{\lambda} C1(t_{i,1}) - \frac{1}{\lambda} t_c \, , C1(t_{i,4}'^{C2}) = \frac{1}{\lambda} C1(t_{i,4}) - \frac{1}{\lambda} t_c \quad (25)$$

Where $C2(t_{i,2}'^{C1}), C2(t_{i,3}'^{C1})$ are timestamps of clock C2 relative to timestamps of clock C1, and $C1(t_{i,1}'^{C2}), C1(t_{i,4}'^{C2})$ are timestamps of clock C1 relative to timestamps of clock C2. So the four timestamps of clock C1 ($C1(t_{i,1}), C2(t_{i,2}'^{C1}), C2(t_{i,3}'^{C1}), C1(t_{i,4})$) and the four timestamps of clock C2 ($C1(t_{i,1}'^{C2}), C2(t_{i,2}), C2(t_{i,3}), C1(t_{i,4}'^{C2})$) can be obtained. Based on the two groups of timestamps, one-way delay between computer C1 and computer C2 can be measured by the equation (26) or (27).

$$d_{C1->C2} = C2(t_{i,2}) - C1(t_{i,1}'^{C2}) = C2(t_{i,2}'^{C1}) - C1(t_{i,1}) \quad (26)$$

$$d_{C2->C1} = C1(t_{i,4}) - C2(t_{i,3}'^{C1}) = C1(t_{i,4}'^{C2}) - C2(t_{i,3}) \quad (27)$$

The measurements of one-way delay between pcamsf0 and amsseu can be showed in figure 7.

The group of data (Traffic3 data) needs pass through 17 routers, and whose least one-way delay is about 210ms. There are 1000 timestamp data in the group that are showed in figure 7. Figure 7 shows that the clock C1 is quicker than clock C2about 0.000125s/s. The relative time relationship between pcamsf0 and amsseu is showed in figure 7, and its relative time model is equation (26) as following.

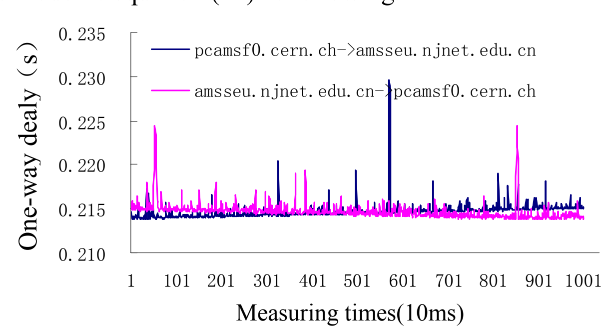


Fig 7 one-way delay between pcamsf0 and amsseu

$$t'_{amsseu} = 1.000125 t'_{pcamsf0} + 216.776019 \qquad (28)$$

Where $t'_{amsseu}, t'_{pcamsf0}$ are the relative time that compares with 2003/6/30 15:00:00. $R^2 = 1.000000$, so the relative time model between pcamsf0 and amsseu can be simulated by linear model fully. The equation (28) also shows that the pcamsf0 clock is quicker than amsseu clock about 0.000125s/s. The figure 8 corrects one-way delay between pcamseu and amsseu using equation (24).

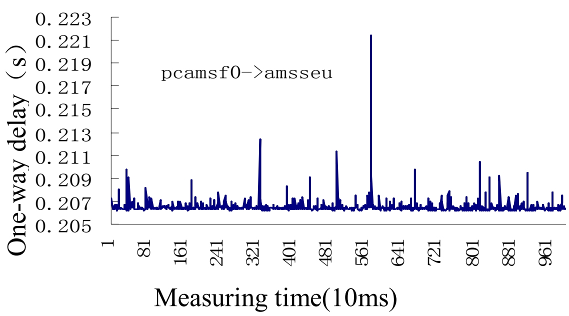


Fig. 8 corrected one-way delay between pcamseu and amsseu

6 Conclusion

When two clocks involved in measurement of one-way delay run at different frequencies and time, inaccuracies are introduced into one-way delay. In the paper, we give three groups of measurement data, which shows that the relative time changes about 0.1ms for per second. That is to say the relative time changing 30ms over the duration of 5minutes at the receiver. It is significant enough to distort one-way delay performance metrics. Instead, the linear increase or decrease in measurements of relative time attests to a constant speed difference between the sender and receiver clocks.

In this paper, we analysis the relative skews and offsets between computer clocks, and establish a relative time model equation to assure time synchronization between two clocks. We measure three groups of exchanging information data between sender and receiver, whose two groups of data are measured inside the njnet.edu.cn network, and one group of data comes from the AMS program that the data is measured between njnet.edu.cn in China and cern.ch in Switzerland. The three groups of data are used to verify the model performance, as well as its estimating error. Last we rely on the relative time model to build correcting one-way delay equation, and the AMS project data is used to experience the correcting one-way delay model. The results show that the estimate of the relative time model is likely to be unbiased and have less variance. In conclusion, the relative time model is simple, fast, and robust.

7 Acknowledgments

This research was partially supported by 863 program of China under grant No. 2001AA112060, Project 90104031 supported by NSFC, and 301 program of Southeast University under grant No. 3009001112.

Reference

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

[2] D. Mills, "Simple Network Time Protocol (SNTP)", RFC 1769, March 1995.

[3] G. Almes, S. Kalidindi, M. Zekauskas, "A One-way Delay Metric for IPPM", September 1999.

[4] K.C Claffy, G. C. Polyzos, and H-W. Braun, Measurement considerations for assessing unidirectional latencies, Internetworking: Research and Experience, Vol. 4, No. 3, pp: 121-132, 1993.

[5] A. Pasztor and D. Veitch, A Precision infrastructure for active probing, in Proceedings of the PAM2001 workshop on Passive and Active Measurements, Amsterdam, Apr. 2001.

[6] Tanja Zseby: Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements, Proceedings of Passive and Active Measurement Workshop 2002, Fort Collins, CO, USA, March 25-26, 2002.

[7] S.B. Moon, P.Skelly, and D. Towsley, Estimation and removal of clock skew from network delay measurements, in Proceedings of INET 2000, Tokyo, Japan, July 2000.

[8] V. Paxson, On calibrating measurements of packet transit times, in Proceedings of ACM SIGMETRICS, Madison, WI, June 1998, pp. 11-21.

[9] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics", Ph.D. dissertation, U.C. Berkeley, 1997.

[10] Cheng guang, Research on Traffic Sampling Measurement and Behavior Analysis in Large-Scale High Speed IP Networks, Ph.D. dissertation Southeast University, 2003, pp: 80-83

[11] http://accms04.physik.rwth-aachen.de/~ams/, 2003.6.30.

[12] http://www.grid.unina.it/grid/mtools/, Mtools