Chapter 1

Introduction

There are applications which require distributed clock synchronization over packet-switched networks. For example, multimedia applications which process time-sensitive information such as voice, audio and moving pictures and also the applications of networked measurements. Since all the concerns related to timing are encompassed by the term “synchronization”, the accuracy of timing is a critical issue, even only few milliseconds of timing deviation may cause fatal errors.

Compared with synchronization over circuit-switched network, packet-switched has advantages like scalability and cost effectiveness. These advantages make the use of synchronization over packet-switched networks keep growing. But satisfying the need of synchronization over packet-switched networks will face the difficulties due to the asynchronous nature of packet-switched networks which may degrade the performance. Thus, the solution adopted for the circuit-switched networks will not scale to the packet-switched networks.

Before achieving synchronization, it is necessary to know the clock skew, i.e., how much the difference is between the frequencies of the two clocks to be synchronized. This difference is to be compensated in order to make the clocks achieving synchronization. To estimate this amount accurately is the aim of the synchronization algorithm.

End-to-end delay measurements are frequently used in analyzing network performance. These measurement data can be a great help in decision making such as optimizing the network resource allocation with the aid of the information of current network status. An one-way measured delay can be obtained by a pair of timestamps: the sender adds a timestamp to the packet at the time when the packet departs from the sender, another timestamp is added by the receiver at the time when the packet arrivals at the receiver, and the difference between the receiver's and the sender's timestamp is called one-way measured delay. Since the measured delays are calculated based on a pair of timestamps according to two different clocks at both end-systems, the information of clock skew can be gathered from these measurements when the two clocks are not perfectly synchronized.

Fig.1 and Fig. 2 are two samples of one-way delay measurements. We can see from these two figures: there's an increasing or decreasing trend in measured delays over the observation duration.

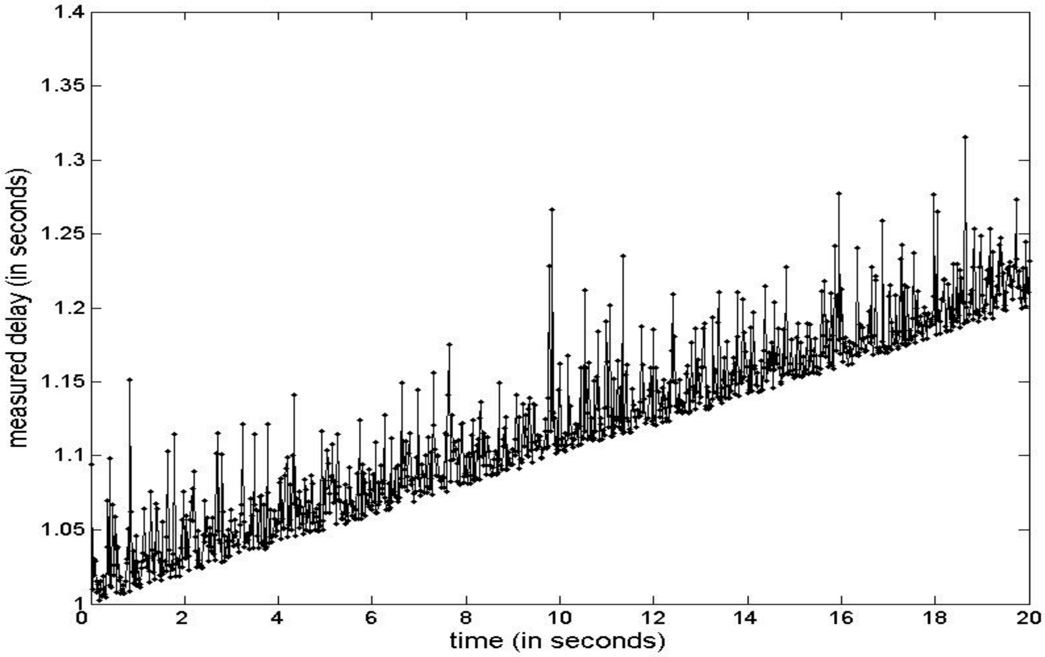

Figure 1 One-way delay measurements with an increasing trend (generated)

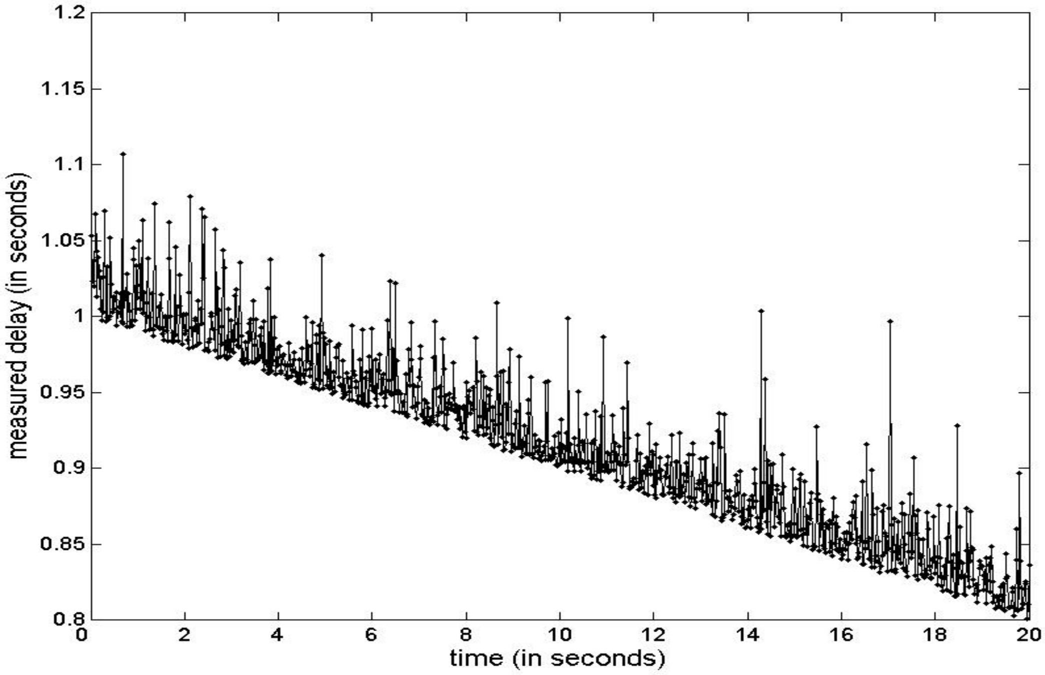

Figure 2 One-way delay measurements with a decreasing trend (generated)

When the network gets congested, there will be an increasing in the range of measured delays, which means the difference between the maximum and minimum observed delay will become greater rather than an increasing trend in the minimum observed delay as in fig.1 [1], and a decreasing trend in the minimum observed delay as fig.2 shows which is hardly a result of network traffic change.

Therefore, we recognize this linear increasing or decreasing trend in the measured delays as a constant frequency difference (clock skew) between the two clocks at end-systems. The goal of the proposed synchronization algorithm is to estimate the clock skew from the increasing or decreasing trend of the measured delays.

In this thesis, we propose a simple clock synchronization algorithm based on the linear least squares method, called iterative linear least squares (ILLS) which can be used both offline and online. Computer simulations are conducted to evaluate the performance of the proposed ILLS algorithm. The rest of this thesis is organized as follows:

In Chapter 2, we introduce the basis of clock synchronization algorithm with a short review of the available literatures concern the clock skew estimation and show the idea behind our proposed ILLS algorithm.

In Chapter 3, we first introduce terms and notations used for describing the clocks behavior and the algorithm and then we describe the ILLS algorithm for both online and offline clock synchronization in detail.

In Chapter 4, we present the results of computer simulations and analyzing the performance of the proposed algorithm. Then we conclude this thesis in Chapter 5.

Chapter 2

Basis of Clock Synchronization Algorithm

Several literatures have discussed the problem of clock synchronization in recent years. The desirable properties that a clock synchronization algorithm should exhibit are fast and robust. Fast means less time is needed to perform the algorithm; and the performance of algorithm is not affected by the magnitude of the actual skew is called robust. These two properties can be used as a basis to judge the performance of an algorithm.

Most of the proposed algorithms in the previous works put their attention on estimating and removing the clock skew while the clock offset is not concerned. There are two reasons for focusing on clock skew: The first one is once the clock skew is accurately estimated, the clock offset is relative simpler to be obtained. And the second one is many applications can only see one-way delay measurements to deal with the synchronization problem, but there is not enough information to distinguish the clock offset and the fixed portion of the end-to-end delays[3]. Thus the clock skew is the main topic of the following discussion.

2.1 Related works

Paxson [2] proposed an algorithm based on median line fitting technique which using the median of the measurements to estimate a line from the increasing trend of the measured delays, and the created line's slope is considered to be the clock skew. But the algorithm has been tested in [3] which performs poor when the data is highly variable.

In [3], the authors proposed a linear programming algorithm for clock skew estimation. They arrange the relationship between the packet's arrival and departure interval to formulate the estimation problem as a linear problem. The goal of the algorithm is to find a line under all the data points and is closest to them as possible. "Closest" here means to minimize the sum of the vertical distance between the line and all the data points. And the slope of the created line is recognized as the clock skew.

In [4], the authors also formulate the estimation problem as a linear problem. The objective function of the optimization problem is to minimize the area between the curve created by the data point and the sought line. The curve is found using a convex hull algorithm. The case with the clock resets is also concerned in this literature which is beyond our scope.

In [5], the authors proposed two solutions to the problem. One is based on the average of the packet's arrival and departure interval to estimate the clock skew. The other one is proposed with an additional assumption: take the effect of the clocks' resolution into account of the problem. In the literature, this assumption limits the accuracy of the measured delays to a few milliseconds or less. Thus the propose algorithm can only obtain the clock skew in units (the clock's resolution).

For online skew estimation and removal, fewer works have been done compared with offline.

In [4], the authors use their convex hull based algorithm to estimate the skew at fixed intervals and to use the last estimate to remove the skew effect from upcoming measures.

In [5], the authors proposed a new technique called sliding window which is fast to respond to skew effect but less accurate than the convex hull algorithm. And the authors combine the two techniques: use the sliding window algorithm during the first interval and to use the convex hull algorithm during subsequent ones to benefit from the advantages of the two approaches.

2.2 Idea behind the proposed algorithm

The method of the linear least squares is a common data fitting technique. It can be used to find the approximated solution of an over-determined system which is a general case in skew estimation problems. But the linear regression algorithm based on the linear least squares method is considered to perform poorly in skew estimation problem due to the nature of the delay measurements [3], [4], [5].

In our opinion, this is because the linear least squares method gives all the data points the same weight in skew estimation, which may works well when the data points are normal distribution. But the network delay measurements rarely satisfy this condition. Networked measurements are more reasonable to have an exponential distribution. That is to say, the data points with variable delays will hide the linear trend in the measured delays, which is contributed from the clock skew and thus degrade the accuracy of estimation when using the linear least squares method.

Here we do some modify in the linear regression algorithm to improve its performance. For eliminating the effect of the variable delay as possible, we apply the linear least squares approach iteratively in order to filter out the data points with greater delays. The procedure is executing until the there is at most one point remaining not filtered out and called the ILLS (iterative linear least squares) algorithm.

In other words, our proposed algorithm gives the data points different weights in the executing procedure; the data points which are filtered out in the iterations can be regarded as having a smaller weight in estimating clock skew. The effect of variable delays is reduced in the iteration procedure.

Thus the idea behind the proposed ILLS algorithm is applying the linear least squares approach iteratively to select the data points which have minimum delay (queuing delay and transmission delay near zero) or close to, and use the selected points which are free from variable delays to estimate the clock skew accurately.

In Chapter 5, the result of simulations will show that the proposed ILLS algorithm exhibits the desirable properties that we mentioned in the beginning of this chapter and have a good performance in estimation accuracy. The detail of the proposed algorithm will be explained in detail in the following chapters.

Chapter 3

The ILLS Algorithm for Clock Skew Estimation

In Chapter 2, we have briefly reviewed the previous work about the problem of clock synchronization and the algorithm for estimating the clock skew. We also have showed the idea behind our proposed algorithm, the concept of how to deal with the variable delays when using the linear least squares method to estimate clock skew. Now we begin to describe the proposed ILLS algorithm and its background.

We will first introduce the terminology and background of the ILLS algorithm in Section 3.1. The detail of the ILLS algorithm will be presented in Section 3.2.

3.1 Terminology and Background

In this section we introduce the terminology and notation for the remaining chapter and formulize the relationship between the clocks at two end-systems as following:

- N : number of packets for analyzing in the algorithm.
- C_s : the sender clock.
- C_r : the receiver clock.
- l_i : timestamp of the i-th packet leaving the sender according to C_s, i = 1, 2, …, N.
- r_i : timestamp of the i-th packet arriving the receiver according to C_r, i = 1, 2,…, N.
- d_i : end-to-end delay experienced by the i-th packet according to C_r, i = 1, 2, …, N.
- a : the clock skew ratio between the two clocks, here we refer $T^s = a \cdot T^r$ for the same duration.
- b : the offset between the two clocks at beginning according to C_r.

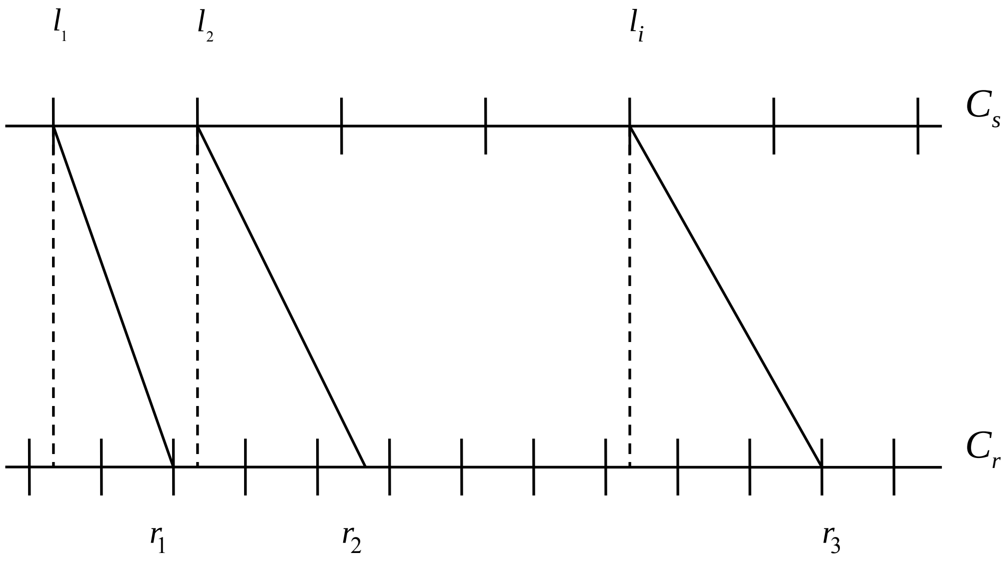

Figure 3 Timing chart showing constant delay

Fig. 3 shows the timing relationship between the sender's and receiver's timestamps based on different clocks at end-system when end-to-end delay is constant. This figure illustrates that when the two clocks at end-systems running at different frequency, the measure delay (the difference between the two timestamps) will not be consistent with the actual end-to-end delay.

The relationship between the two timestamp can be formulated as:

$$r_i = a \cdot l_i + d_i + b \ , \qquad 1 \le i \le N \tag{1}$$

The measured delay of the i-th packet from the difference of the two timestamp therefore is:

$$\begin{aligned} r_i - l_i &= a \cdot l_i + d_i + b - l_i \\ &= (a-1) \cdot l_i + d_i + b \;, \qquad 1 \le i \le N \end{aligned} \tag{2}$$

From (2), if the two clocks at the end-systems are not perfectly synchronized, which means a is not equal to 1, the term $(a-1)\cdot l_i$ and b will contribute extra amounts to the real delay. And the term $(a-1)\cdot l_i$ will increase or decrease as the time progress which lead to a increasing or decreasing trend in the measured delay as in fig.1 and fig.2.

Thus the objective of a skew estimation algorithm is to estimate the clock skew a when d_i is unknown. N sets of the two timestamps l_i and r_i are the only information can be obtained.

3.2 The ILLS algorithm

There are N sets of timestamps which are N sets of equations for the algorithm to estimate the clock skew as in (1)

$$r_i = a \cdot l_i + d_i + b \ , \qquad 1 \le i \le N$$

Subtracting $r_1 = a \cdot l_1 + d_1 + b$ from (1), in order to eliminate the offset term b, and we get:

$$r_i - r_1 = a(l_i - l_1) + d_i - d_1 \ , \qquad 2 \le i \le N \qquad (3)$$

Rearranging (3) into an matrix form to obtain

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{b} + \boldsymbol{\varepsilon} \qquad (4)$$

Where

$$\mathbf{A} = \begin{bmatrix} l_2 - l_1 & -1 \\ \vdots & \vdots \\ l_N - l_1 & -1 \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} a \\ d_1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} r_2 - r_1 \\ \vdots \\ r_N - r_1 \end{bmatrix}, \ \boldsymbol{\varepsilon} = \begin{bmatrix} d_2 \\ \vdots \\ d_N \end{bmatrix}$$

When $\boldsymbol{\varepsilon}$ is smaller compared to other quantities (general case in skew estimation problem) in (4), an approximate solution of the over-determined system can be obtained by using the linear least squares method:

$$\mathbf{X} \simeq (\mathbf{A}^{\mathbf{T}} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^{\mathbf{T}} \cdot \mathbf{b} \tag{5}$$

Where $\mathbf{X}(1) = a$ is the estimated clock skew.

In (5), since the term $(\mathbf{A}^{\mathbf{T}} \cdot \mathbf{A})^{\mathbf{-1}}$ is a 2x2 matrix, it's easy to find its inverse for further calculation. Therefore, it's time-efficient to execute the algorithm.

The following is the procedure of applying the proposed ILLS algorithm for the clock skew estimation:

- Step1: Set all the data points to the valid set.
- Step2: Use linear least squares method to fit a line to the data points in the valid set.
- Step3: Remove the points which are above the created line from the valid set.
- Step4: Use linear least squares method to fit a line until there is at most 1 point in the valid set.
- Step5: The last created line's slope is the estimated clock skew.

Fig.4 and 5 illustrate step2 and 3 of the proposed algorithm.

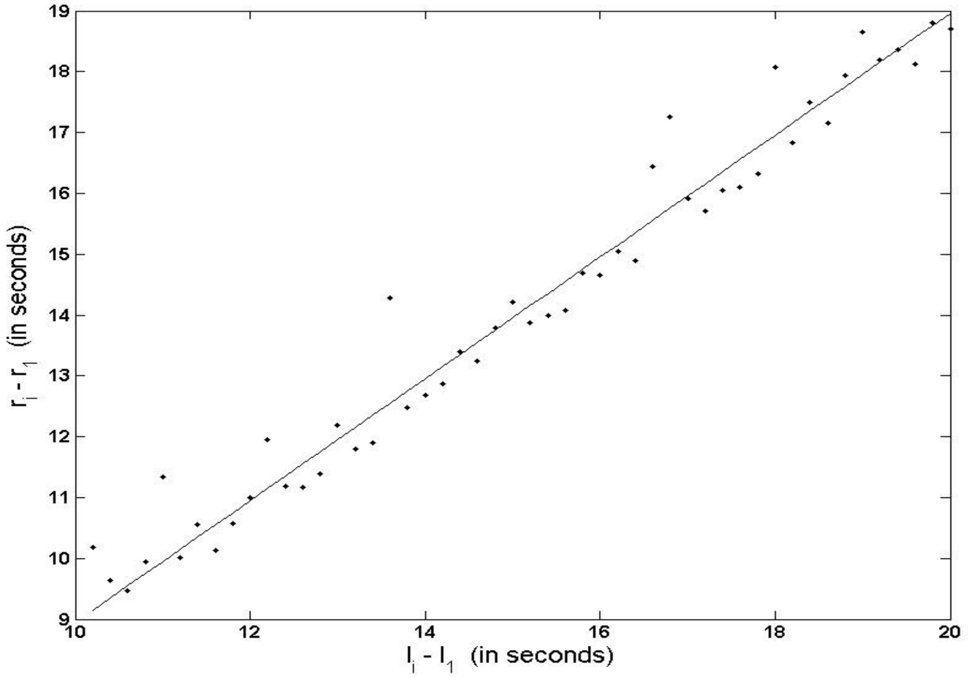

Figure 4 Use linear least squares method to fit a line to the data points in the valid set

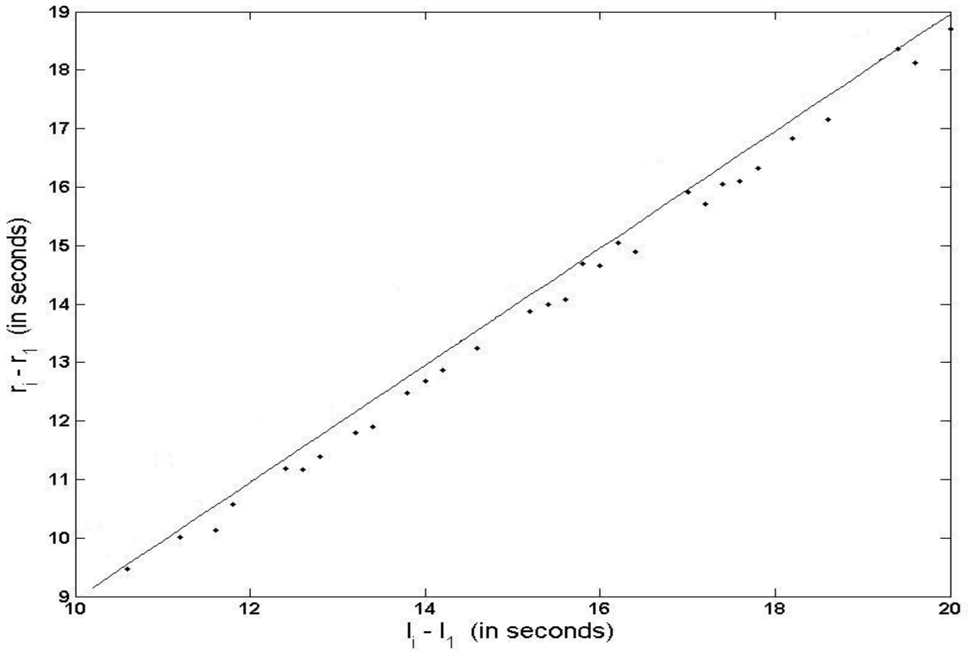

Figure 5 Removing data points which are above created line from the valid set

For estimating the clock skew online, we want to keep tracking the skew for a long period of time and obtain the current best skew estimate in each interval. The ILLS algorithm can be applied interval by interval and an exponential smoothing technique is used to avoid the estimate changing too abruptly.

$$\hat{\mathbf{X}}_{\mathbf{i}} = \alpha \cdot \hat{\mathbf{X}}_{\mathbf{i\text{-}1}} + (1-\alpha) \cdot \mathbf{X}_{\mathbf{i}} \tag{6}$$

In (6), $\hat{\mathbf{X}}_{\mathbf{i}}$ and $\hat{\mathbf{X}}_{\mathbf{i\text{-}1}}$ are the results of the skew estimate in the current and the previous interval, respectively. $\mathbf{X}_{\mathbf{i}}$ is the output of the ILLS algorithm in the current interval where α is the weighting factor.

Thus the obtained skew estimate of interval i is based on the most recent estimation result (from the previous interval) and the most recent observation (the output of the algorithm). We will verify if the ILLS algorithm can track the clock skew in a long period of time in Chapter 5.

Chapter 4

Simulations

In this chapter, we demonstrate computer simulations to verify the proposed ILLS algorithm. This chapter is organized as follows. In Section 4.1 we will show the accuracy of the ILLS algorithm when we use it to estimate the clock skew offline. In Section 4.2, we test if the ILLS algorithm is robust. The performance comparision between the propose ILLS algorithm and the available linear programming algorithm is given in Section 4.3. In Section 4.4, we apply the ILLS algorithm to estimate the clock skew online to test if the algorithm can track the skew for a long period of time

All the computer simulations are conducted by using Matlab 7.01 on the Windows XP platform with Intel Pentium 4 processor at 3.0 GHz and 2.0 Gb RAM.

4.1 The accuracy of the ILLS algorithm

In this section, we test our ILLS algorithm with different delay distribution. The assumption of simulations are made as follows: exponential distributed end-to-end delay with a mean 2, 20 and 200msec according to the receiver's clock, respectively; the packet's inter-departure time is 200msec according the sender's clock; the number of packet for analyzing is 1000; the clock skew ratio (receiver to sender) is 1.001, which means the receiver's clock will tick fast 1sec than the sender's after 1000sec according to the sender's clock.

Table 1 is the result of the average estimation accuracy and the worst case of the ILLS algorithm with different delay distribution from 100 run. We can see from the table, the average accuracy of the proposed algorithm have a satisfactory performance in all scenario. Even the worst case in the 100 run, the accuracy is still acceptable.

TABLE I. ACCURACY OF THE ILLS ALGORITHM (DIFFENENT DELAY DISTRIBUTION)

	Mean Delay		
	2msec	***20msec***	***200msec***
Max Error (in skew ratio)	**4.6128e-007**	**2.8104e-005**	**8.3180e-004**
Average Error (in skew ratio)	**5.6270e-008**	**7.9595e-007**	**1.6301e-005**

4.2 The property of robustness

In this section, simulation is conducted to verify if the ILLS algorithm is robust. The tested clock skew magnitude is 1.001, 1.01, and 0.999, respectively. End-to-end delay has an exponential distribution with a mean of 20msec. The other assumptions are consistent with the first simulation in Section 4.1.

Table 2 is the result of the average estimation accuracy of the ILLS algorithm with different skew magnitude from 100 run. We can see from the table, the performance of the proposed ILLS algorithm is not affected by the magnitude of the actual skew which means the ILLS algorithm has the property of robust.

TABLE II. ACCURACY OF THE ILLS ALGORITHM (DIFFENENT ACTUAL SKEW MAGNITUDE)

	Actual Skew Magnitude		
	1.001	***1.01***	***0.999***
Average Error (in skew ratio)	**7.7847e-007**	**7.7847e-007**	**7.7847e-007**

4.3 Comparison between the ILLS and linear programming algorithm

In this section, we compare the performance of the ILLS algorithm and the linear programming algorithm proposed in [3]. The assumptions of simulation are made as: exponential distributed end-to-end delay with a mean 2, 20 and 200msec according to the receiver's clock, respectively. The packet's inter-departure time now is 20msec according the sender's clock. Other assumption is made consistent with the first simulation in Section 4.1

Table 3 shows the average estimation accuracy of the ILLS and the linear programming algorithm with different delay distribution from 100 run. We can see from the table, the performance of the ILLS algorithm has an obvious improvement in all scenario compared with the linear programming algorithm proposed in [3].

Table 4 shows the average execution time needed to perform the ILLS and the linear programming algorithm, respectively. Compared with the linear programming algorithm, the ILLS algorithm is much more time-efficient in execution.

TABLE III. AVERAGE ERROR (IN SKEW RATIO) OF THE ILLS AND LINEAR PROGRAMMING ALGORITHM

	Mean Delay		
	10msec	***20msec***	***200msec***
ILLS	**1.9833e-006**	**5.7029e-006**	**5.972e-005**
Linear Programming	**4.9133e-005**	**6.2314-005**	**1.5292e-004**

TABLE IV. AVERAGE EXECUTING TIME OF THE ILLS AND LINEAR PROGRAMMING ALGORITHM

	Algorithm	
	ILLS	***Linear Programming***
Average Executing Time	**5.94e-003 sec**	**8.31e-001 sec**

4.4 Tracking ability of the ILLS algorithm

In this section, we demonstrate a simulation to test if the proposed ILLS algorithm can track the clock skew accurately when the skew is not constant in the estimation duration. The assumptions are made as follows: an exponential distributed end-to-end delay according to the receiver's clock; and the mean delay is 20msec in each estimation intervals, the packet's inter-departure time is 200msec according to the sender's clock. The ILLS algorithm is applied interval by interval (100 packets in one interval) with a total 1500 packets in the estimation duration. The clock skew ratio is varied at the 5th and 10th interval (from 1.05 to 1.002 at the 5th interval and 1.002 to i.001 at the 10th interval). The weighting factor α is set to 0.1. The test is running 100 times.

Fig. 6 and Table 5 are the average estimation error in each interval of the tracking duration. The results show that the proposed ILLS algorithm can respond to the variation of the clock skew quickly. Since in the variation of the clock's frequency is not frequent in a short period of time in real situation, we can say that the ILLS algorithm can success track the clock skew.

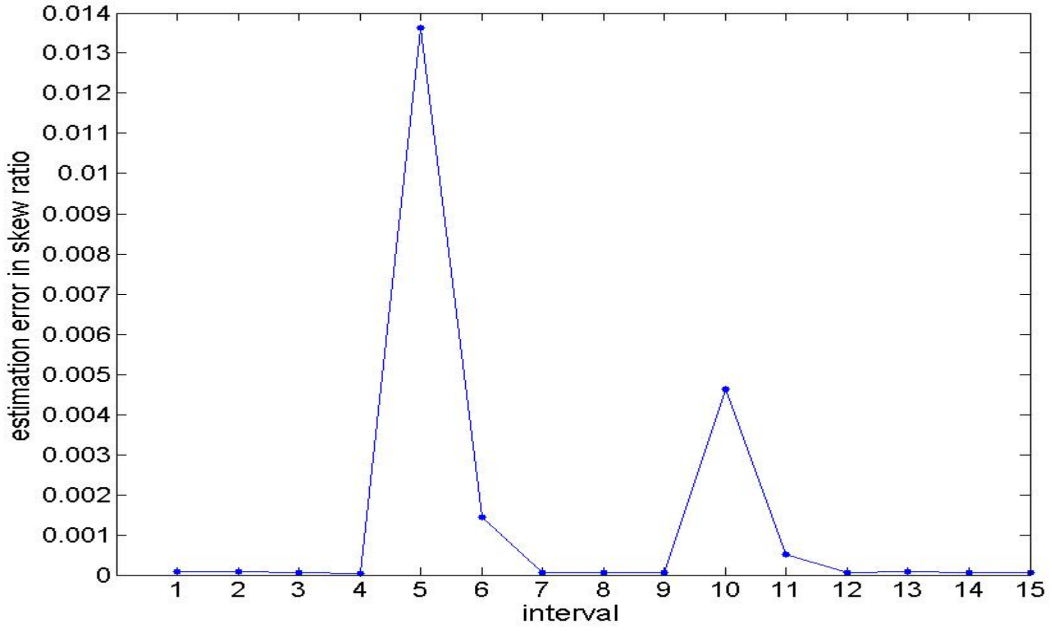

Figure 6 Average error (in skew ratio) in the estimation duration

TABLE V. AVERAGE ERROR (IN SKEW RATIO) IN EACH INTERVAL OF THE ESTIMATION DURATION

Interval	**1**	**2**	**3**	**4**	**5**
Error in Skew Ratio	**7.517e-005**	**8.6353e-005**	**5.8105e-005**	**4.3946e-005**	**0.01363**
Interval	**6**	**7**	**8**	**9**	**10**
Error in Skew Ratio	**0.0014326**	**5.9025e-005**	**5.6673e-005**	**6.6903e-005**	**0.0046205**
Interval	**11**	**12**	**13**	**14**	**15**
Error in Skew Ratio	**0.00052247**	**6.9539e-005**	**7.3315e-005**	**6.2192e-005**	**5.0213e-005**

Chapter 5

Conclusion

In this thesis, we have briefly introduced the framework of the clock synchronization problem for understanding what we deal with. The ILLS algorithm is proposed for clock skew estimation which can be used both offline and online. In skew estimation offline, the simulation results show that the ILLS algorithm has a satisfactorily accuracy in all delay distribution scenario and an obvious improvement in both estimation accuracy and execution time compared with the linear programming algorithm. In estimating the clock skew online, the proposed algorithm can success track the variation of clock skew.

In conclusion, the proposed ILLS algorithm is fast and robust which can be used in a wide range of application.

Bibliography

[1] W.Zhang and J.He, “Study of the Characteristics of End-to-End Delay,” Proceedings of International Conference on Communication Technology, 2006.

[2] V.Paxson, “On Calibrating Measurements of Packet Transit Times,” Proceedings of SIGMETRICS’ 1998.

[3] S.B.Moon, P.Skelly, D.Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements", Proceedings of IEEE INFOcom 1999.

[4] L.Zhang, Z.Liu, C.H.Xia, Clock synchronization algorithms for network measurements, Proceedings of IEEE INFOcom 2002.

[5] H.Khlifi., J.C.Gregoire, “Estimation and removal of clock skew from delay measures” Proceedings of IEEE LCNcom 2004.