RL-TR-94-87
In-House Report
July 1994

GLOBAL TIME TRANSFORMATIONS

Jon B. Valente and Robert J. Vaeth

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-94-87 has been reviewed and is approved for publication.

APPROVED:

ANTHONY F. SNYDER, Chief
C2 Systems Division
Command, Control, and Communications Directorate

FOR THE COMMANDER:

JOHN A. GRANIERO
Chief Scientist
Command, Control, and Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (C3AB) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1994	In-House

4. TITLE AND SUBTITLE
GLOBAL TIME TRANSFORMATIONS

5. FUNDING NUMBERS
PE - 62702F
PR - 5581
TA - 21
WU - AC

6. AUTHOR(S)
Jon B. Valente and Robert J. Vaeth

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3AB)
525 Brooks Road
Griffiss AFB NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-94-87

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3AB)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Jon B. Valente/C3AB (315) 330-3241

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report on global time transformations comes from an in-house research effort in the area of distributed-parallel computing. The distributed-parallel computing environments are multiple clock systems interconnected by communications networks. The global time transformation maps the data from these different systems into a common (i.e., global) clock for an analysis. The derivation of the global time transformation is given. The primary components of the global time transformation are the clock rate and clock offset. These components and the measurement techniques for measuring these components are discussed. To demonstrate the accuracy of the global time transformation, a set of measurements were taken to compare with the transformation predictions.

14. SUBJECT TERMS
Clocks, Time Like Variables, Time Signals, Time Dependence, Time Domain, Computer Networks

15. NUMBER OF PAGES
72

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	U/L

Table of Contents

Preface

As technology advances, more and more computer systems are being interconnected by communication networks so users can share data and applications. Data analysis on a network environment containing multiple computers is very difficult. The global time transformation approach reduces the difficulty in doing the analysis.

This Global Time Transformations report is the product of an in-house research effort in the area of distributed-parallel computing. The effort involves a set of experiments that were executed over a distributed network between Rome Laboratory (RL) in Rome, New York and the Northeast Parallel Architecture Center (NPAC) at Syracuse University in Syracuse, New York. During each of the experiments, data was collected from four separate sources. To do an analysis of the experiment, we had to reconstruct the order of the events from the data. When we tried to bring the separate sets of data together to reconstruct the events during the experiment, we encountered a problem. Because the timestamp data for each system was different, there was no way to relate the data from one system to other system. To bring the data together, a time transformation was derived that would map each system's data to a common time base(i.e. the global time base). The purpose of the Global Time Transformation is to take the *local time* data and convert it to the equivalent *global time* data.

Determining the exact time for any event is physically impossible. The intention of the Global Time Transformation is to map the timestamp datum into datum that is much closer to the actual global time. The accuracy of the transformation will depend on the precision of your global time transformation. The major factors that limit the precision are the resolution of the time measurement capability and observability of the metrics.

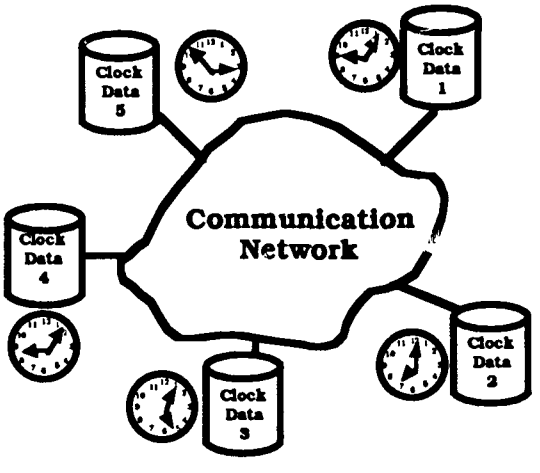
This report describes the Global Time Transformations part of the distributed-parallel in-house effort. The report is divided up into seven sections. Section 1 is the introduction to the report and explains where the global time transformation techniques fit into the general technology base. Section 2 defines the function of the global time transformations

and derives the transformation. Section 3 defines the clock drift rate and discusses the various techniques used to calculate the clock rate. Section 4 defines the clock offset and discusses a couple of the techniques for measuring and calculating the clock offset. Section 5 compares the global time transformation predictions with actual measurements of clock drift. Section 6 recommends candidate areas of follow-on research. Section 7 is the conclusion for the report.

Section 1: Multiple Clock Systems

Introduction

As computers become more prevalent and applications become more complex, it is not surprising that many solutions to today's problems involve multiple computers. As technologies like distributed processing and cluster processing mature, there will be a significant increase in the number of multiple computer systems. Unfortunately, when new technologies provide opportunities to solve problems, new problems are introduced. One of those problems is that a multiple computer system has multiple clocks and each clock has a different time (see Figure 1A). None of these clocks are set to the same time nor do they run at the same speed. An analysis under these conditions is impossible. If the metric for your analysis is time, then analyzing a multiple clock system is a problem. The fundamental problem is that there is no direct way to relate the data from one system to the data from another system.



Multiple Clock Data
Figure 1A

There are many multiple clock systems and situations in which data from multiple clock systems have to be used or analyzed. One sit-

uation is when you try to create order from the data[1]. For example, suppose you wanted to observe the execution of an application for analytical reasons. The analyst runs an experiment and collects several sets of data. Without a relationship to associate the data in one time domain to the data in the other time domain, there is no way to order the events. Any attempts to produce the proper sequence of events will fail. During the ordering process, there are usually some logical sequences of events that can be used to detect the need for a transformation. For example, the arrival time for a message on one system can precede the departure time for a message on the other system. The ordering becomes more difficult when the observer only sees a small part of an event. For example, the Rube Goldberg machine is a machine that has a well defined sequence of events whose goal is to capture a mouse or turn on an appliance. Let two events in a Goldberg's machine sequence be a ball dropping down a chute and the ball releasing a balloon. Suppose there are two observers and each is assigned a different event to watch and record the starting time for the event. One observer saw the ball being dropped down the chute at 12:00 PM his time and another observer saw the ball release a balloon at 11:39 AM his time. How much time was there between the time the ball was dropped until it released the balloon? The time data corresponding to the events can not be used to answer this question without some way of relating the data from one observer to the data of the other.

Merging is another situation where you encounter the multiple clock problem. The motivation for merging is to make one database out of multiple data bases or combine subsets of multiple data bases to form a new data base. When the rules for merging are based on their time values, then the problem surfaces again. If data values in one database cannot be associated with the data in the other, merging of the databases is not possible. Once again, some relationship is required to associate the data in one system to the data in the other system.

Numerical analysis is another type of data processing that encounters the multiple clock problem. Numerical analysis quite often uses data from multiple clocks to do calculations. When you do a perfor-

[1] Since the events can occur concurrently, the ordering really is a partial ordering of the events.

mance analysis on a multiple computer system or distributed system, data for the analysis come from different computers and network analyzers. With data in its' original form, it is impossible to use. To do the analytical calculations, the analysts must either transform the data from one clock into the time base of the other or transform the data from both clocks to a common time base. In either case, there is a need to transform the data.

The multiple clock problem is a very common problem and a solution needs to be found before an analysis can be performed. *There is no **exact** solution to the multiple clock data problem.* The physics of the system (i.e. software and hardware delays and the speed of an electron is finite/bounded) make it impossible to measure the exact time, even if there is only one clock. The measurements will always be approximate values to the exact/actual time.

Hardware Techniques

The solutions to the problem are either to set and synchronize all the clocks so that they will always read the same or to accept the fact that they are not the same and make the proper adjustments to the data. Of those techniques that attempt to synchronize the clocks, the most common techniques are the hardwire connection, the radio clock and the time protocols techniques. The most common technique for adjusting the data is the software data transformation technique.

Of the synchronization techniques, the easiest is the hardwire connection. There are two hardwire techniques. The first hardwire technique is to hardwire all the timestamping mechanisms to the same clock. This would enable the separate computer systems to use a common clock. There are implementation problems, like clock contention, which creates wait periods that impact the timestamp data. There are other problems in hardwiring the system. Most systems cannot be hardwire modified because they are proprietary and the schematics are not available. Chip technology and firmware adds to the difficulty in modifying these systems. Another disadvantage is that most systems usually involve a distributed network, which makes hardwiring not feasible or practical.

Another hardwire technique uses a common hardwire to synchronize the multiple clocks to read the same. Once again, the reasons mentioned in the previous technique are also true for this technique. The advantage is that there is less clock contention. Hardwire techniques are difficult and sometimes impossible to implement, are very expensive and always contain additional hardware delays.

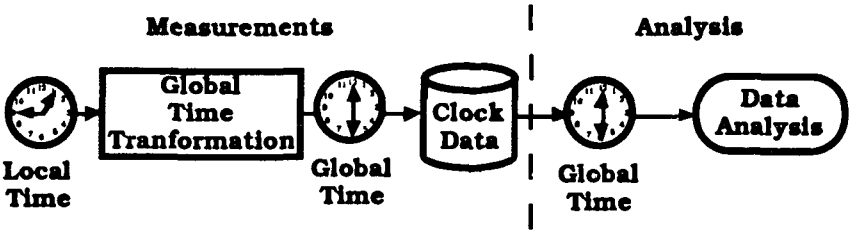
The biggest problem with hardwire connection techniques is that they use wire for connectivity. The hardwire connection technique becomes less feasible and less cost effective when the distance gets greater then a few thousand feet. One way to eliminate the wire is to use a "radio signal". This radio signal technique broadcasts a signal from one location and each of the other systems receives the radio signal. The signal is then used to align its clock. This technique is similar to the US radio station WWV, in Boulder Colorado, that broadcasts a time-hack every minute so that everybody can set their watches and clocks. The advantage of this technique is tha' it doesn't need miles of wire to work. The disadvantages are that it is victim to propagation delays, it requires expensive hardware (i.e. radio equipment and interface boards) and the clock circuitry is not always accessible to do the synchronized.

One of the problems with the radio signal technique is the expensive hardware. The logical thing is to reduce or eliminate the expensive hardware. Let's recall that the operational environment for this effort is a set of computers connected by a communication network. Since the connectivity is already being provided by the network, the network can be used to do the synchronization. To take advantage of the network connectivity, special timing protocols were designed specifically for synchronizing clocks (e.g. the Network Time Protocol (NTP) and the Digital Time Service (DTS)) [Mills 89b]. NTP is a hierarchical structure of primary and secondary servers that work in a client/server arrangement. The client system dynamically interacts with the NTP servers to synchronize the clients clock. DTS is a synchronization subnet which "...consists of clerks, servers, couriers and time providers. With respect to the NTP nomenclature, a time provider is a primary reference source, courier is a secondary server intended to import time from one or more distant primary servers for local redistribution, and a server is intended to provide time for possibly many end nodes or clerks." [Mill89a, Mill89b, Mill92]

The time protocol is less expensive than the hardwire techniques but does have some basic requirements that must be met by the client system. The client system must be able to modify the clock, create and modify messages and interact with the servers. There are two major problems with the timing network protocol approach. One, many systems can't meet all the functional requirements. Two, there is still some error in the clock due to the delays in network communication and processing of the protocols. In the recommended research section of this report, suggestions are made as to a possible technique for reducing the error in the network timing protocol techniques.

Software Techniques

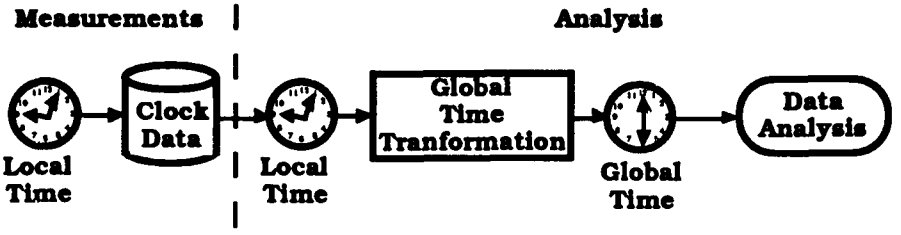
As discussed earlier, sometimes the operational environment makes it impossible to synchronize the clocks and that alternative solutions have to be found. One class of solutions assumes that multiple unsychronized clocks generate the data and that the data needs to be adjusted. The adjustment to the data is done by a data transformation. The data transformation technique maps the data from one time domain to a common time domain so that the data appears to have come from a common clock. These techniques are the software data transformation techniques and are categorized as an on-the-fly technique or an after-the-fact technique or a hybrid technique.



On-the-Fly Technique
Figure 1B

The ***on-the-fly technique*** takes the timestamp datum and transforms it to a global time datum where it is either used immediately or is stored to be used later (see Figure 1B). One advantage of the on-the-fly technique is that the data appears in only one form which requires less storage. Another advantage is that the data is transformed only once

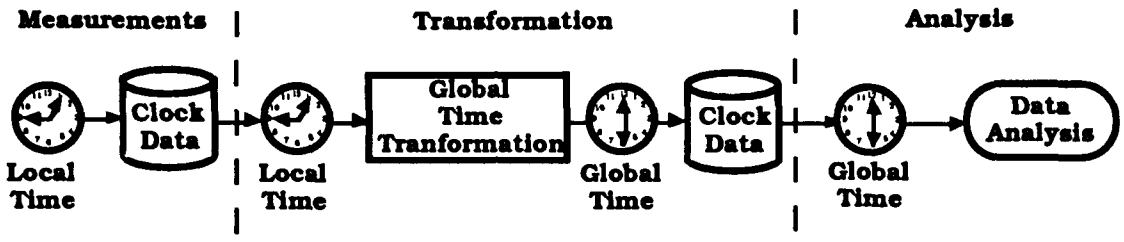
and the analysis program doesn't have to spend time doing the transform. A disadvantage of the technique is that the original data is not stored. If something happens during the transformation, the data is lost. Another disadvantage is that implementing this technique on a uniprocessor system results in a shift in the data measurements. If a uniprocessor is doing the transformation on previously measured data when it is time for a new measurement, the measurement is delayed. This delay produces an error in the measurements. The magnitude of the shift varies in time and depends on the current environment. This could be reduced/eliminated in a multi-tasking environment by having the measurement processing independent of the normal processing.



After-the-Fact Technique
Figure 1C

For the ***after-the-fact technique*** the data is taken, stored and later transformed to global time data and used for the calculations (see Figure 1C). This technique has an advantage that it preserves the original data. This is important if the data is to be used for multiple applications or if a problem occurs in during the transformation. Another advantage of the technique is it delays the transformation until later, so the data measurements are more accurate because there is no contention for the uniprocessor when the measurements are being taken. A disadvantage is that it requires more storage then an on-the-fly technique because the data exists in two forms, the original and transformed form. If the same data is accessed often it would be advantageous to process and store the data in the translated forms. This is done by transforming either the entire set of data or as a subset of the data. Another disadvantage is that the transformation, when done during the analysis, slows down the analysis because it uses some of the processor time. This could be a problem for those analyses which are data intensive.

The best technique is the ***hybrid technique*** which gives the analyst the best of both techniques, even on a uniprocessor system (see Figure 1D). The basic difference is that this technique takes three steps to complete while the previous techniques use only two steps. Step one, the data is measured and stored during the measurements part of the experiment like the after-the-fact technique. Therefore, the measured



Hybrid Technique
Figure 1D

data is not effected because the processor is dedicated to the measurement process. Step two, after all the measured data has been taken, the time transformation maps the data into the common time value which is stored into permanent storage. Step three, the *transformed data* is then used to do the data analysis.

The additional step is step two where the technique preprocesses the data before the analysis. One advantage is that the measured data is more precise because the data is not affected by the delay generated by the transformation processing. Another advantage is that the data are stored in both the original and transformed forms. The technique is more reliable because the error recovery possibilities are better. Another benefit is that during the data analysis there is no processing time required to transform the data. This time savings speeds up the analysis. Even more time is saved when multiple requests are made for the same data entry. The main disadvantage is that the data must be stored in both forms at the same time. If the set of data is large, then the final storage requirement is very large.

This section discussed some of the primary solutions to the multiple clock data problem along with the advantages and disadvantages. From the list of candidate solutions, the software solutions are the most feasible, flexible, and cost effective. A close look at the software techniques

reveals that each technique involves a data transformation. This report deals with the software data transformation called the Global Time Transformation.

Section 2: Global Time Transformation

The Perfect Clock

The ideal situation for data analysis is when the time data from each clock in a multiclock system always reads exactly the same as every other clock in the system. For all these clocks to always read the same, they must all be set to the same time, must always run at the same speed and the speed must be constant. Clocks that can do this are "perfect clocks."

A formal definition of a ***perfect clock*** is a clock that is correct, accurate, and stable [Duda 87]. A clock is "***correct***" if the reading on the clock is the same as the reference time. A clock is "***accurate***" if the speed of the clock is the same as the speed of the reference clock. This means that a second on one clock is the same as a second on the other clock. The speed of the clock can be determined by examining the derivative of the clock readings with respect to time. At those times where the derivative is 1 the clock is accurate[2] . A clock is "***stable***" if the difference in the speeds between the clock and the reference clock is a zero. The stability of the clock can be determined by examining the derivative of the clock speed with respect to time. At those times where the difference is 0, the clock is stable. For a clock to be a "perfect" clock it must always read the same time as the reference clock, must always change time at the same rate as the reference clock and the clock rate never changes.

The global time transformation is a transformation which maps local clock times into global (i.e. reference) clock times (see figure 2A). The global time transformation expressed as a polynomial of infinite order would look like

$$G(t_{Local}) = a_0 + a_1 \cdot t_{Local} + a_2 \cdot t_{Local}^2 + a_3 \cdot t_{Local}^3 + a_4 \cdot t_{Local}^4 + a_5 \cdot t_{Local}^5 + \ldots$$

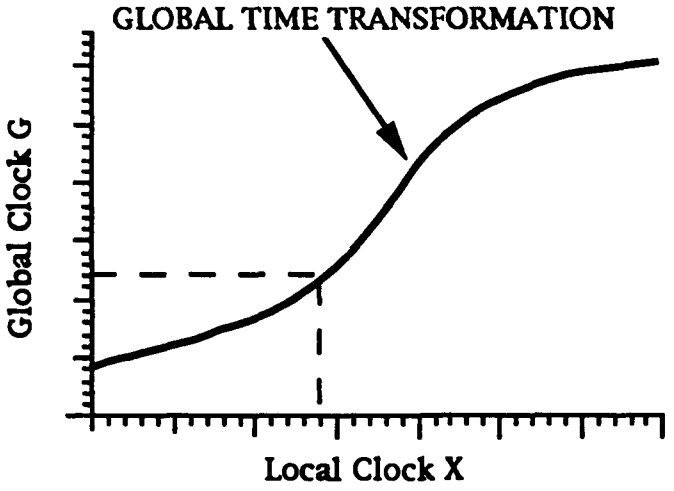
---

[2] Note: A clock being accurate does not mean that the clock reading and the reference time are the same. For both clocks to have the same reading the clocks have to be correct.

For a perfect clock, the global time transformation would be

$$G(t_{Local}) = 0 + (1) \cdot t_{Local} + (0) \cdot t_{Local}^2$$
$$= t_{Local}$$

and the higher order coefficients are 0. A clock is perfect when the local time(i.e. t_{Local}) is equal to the global time (i.e. $G(t_{Local}) = t_{Local}$). Unfortunately, the clocks that are in the multiple clock systems are not perfect clocks. So, the coefficients of the polynomial expression for real clocks are $a_0 \neq 0$, $a_1 \neq 1$ and $a_2 \neq 0$ and the higher order coefficients are not always 0.



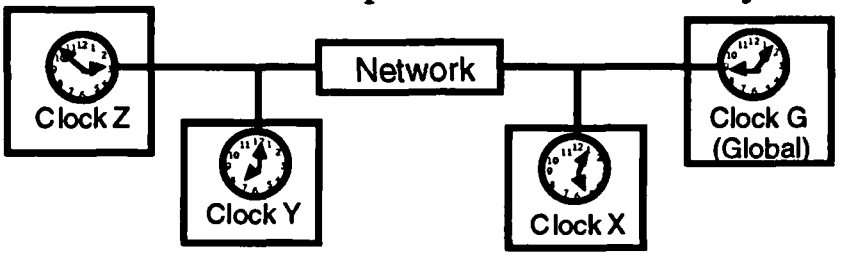
General Global Time Transformation
Figure 2A

The perfect clock definition and transformation gives some insight into what are the key characteristics of clocks. The clock closest to being a perfect clock is the atomic clock. The atomic clock still has to have periodic adjustments made to it to compensate for the fact that it is not a perfect clock. Since the system's clock is not a perfect clock, it drifts relative to the global clock and obviously requires adjusting to keep it close to being correct, accurate and stable.

A Typical System

The next step is to find out how perfect is a typical system clock. To examine the clock behavior, we implemented a set of experiments which

enabled us to observe some electronic clocks, record their times and analyze their time data. Figure 2B is a diagram of the hardware configuration for the experiment. Clock G is the reference clock and will be the clock that all other times will be compared with and mapped for this report. The software controlled the experiment and the message timing, recorded and stored the timestamp data and did the analysis.



Configuration of the Experiment
Figure 2B

In the experiment that was implemented, clocks X and Y were LAN protocol analyzers each having an internal clock. Systems Z and G are parallel computers. The experiment was simple. At hour intervals, systems X, Y and Z send a message to system G. System G receives, timestamps and stores the data . After completing the experiments, the observed differences between clocks are calculated. The results of the calculation are shown in Figures 2Ca, 2Cb and 2Cc. The plot is a locus of boxes, †,

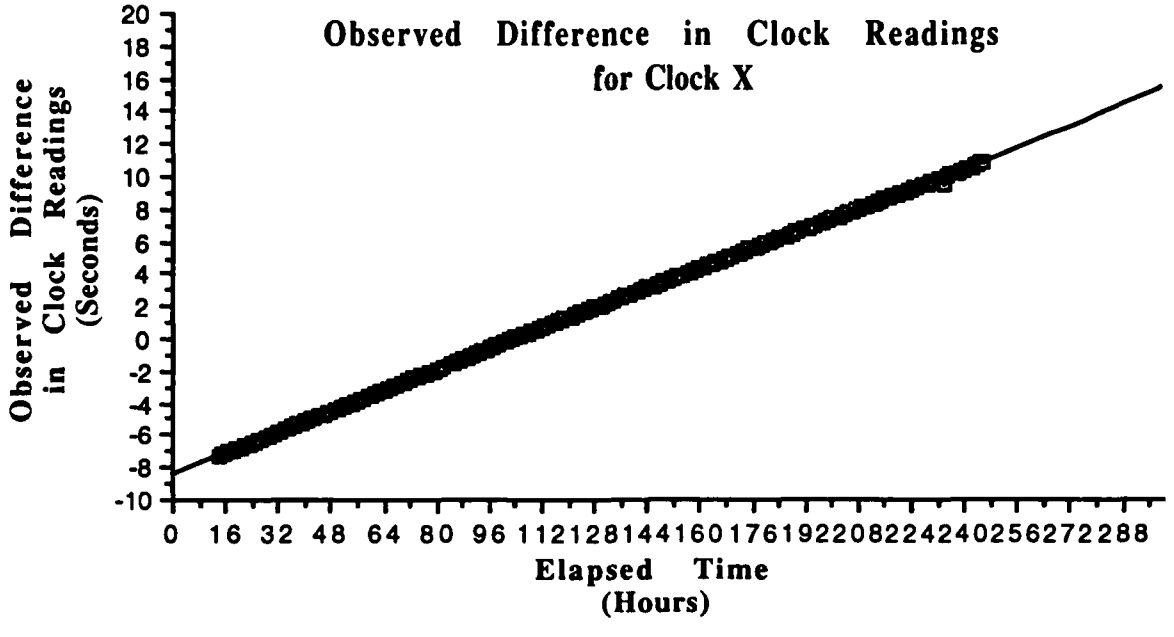


Figure 2Ca

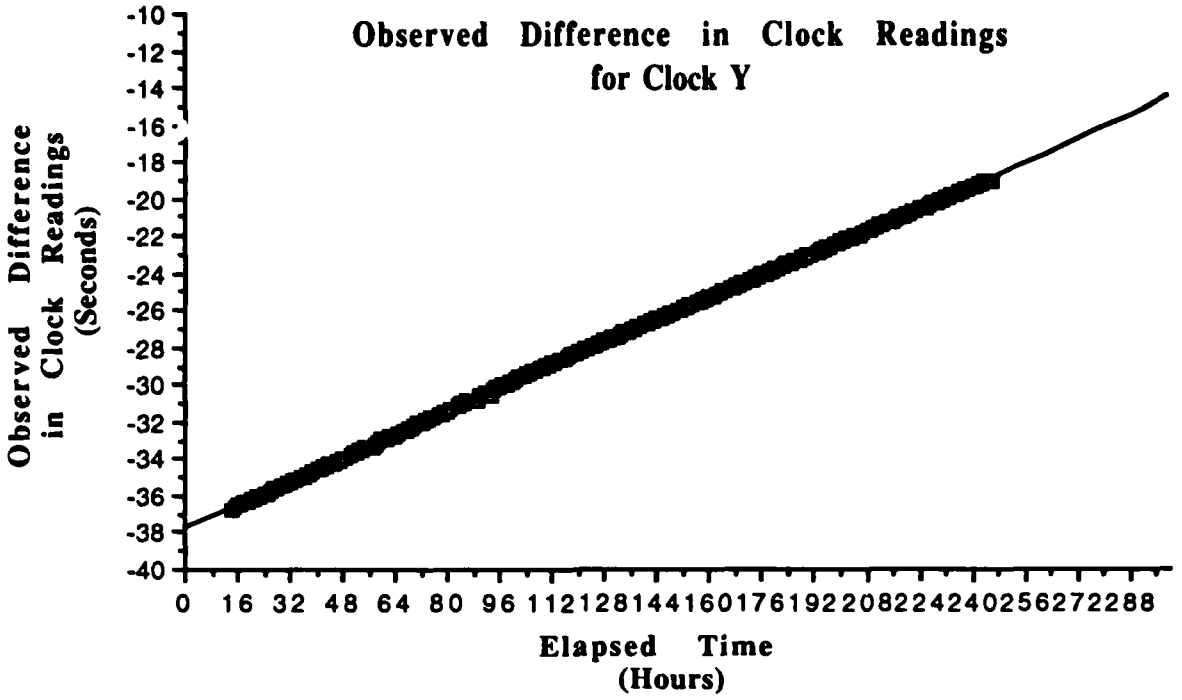


Figure 2Cb

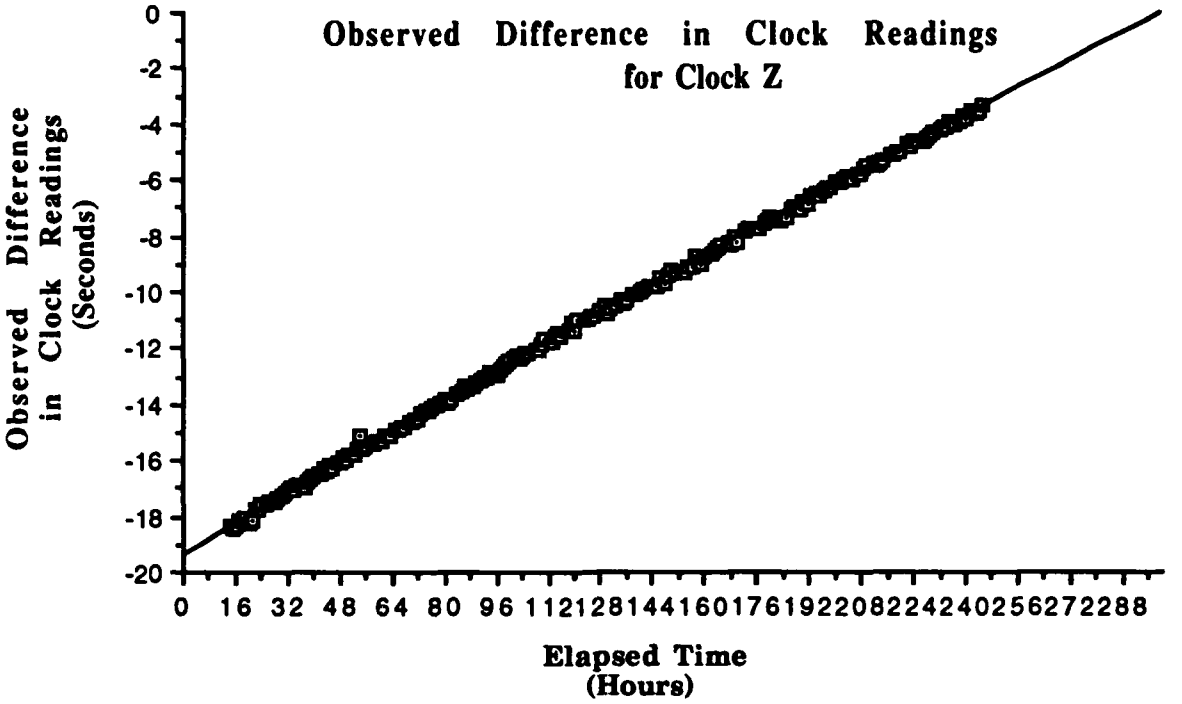


Figure 2Cc

which mark the data points and a linear approximation to the data points.

All three of these plots have three fundamental characteristics.

1) each clock was not initially set to the same time as clock G,
2) each clock drifts at a different rate and
3) each plot appears to be linear.

To develop an understanding for the characteristics of the data, we curve fit each of the data sets and looked at the coefficients. The results of the curve fitting are in the following table:

	Clock X	Clock Y	Clock Z
a_0	8.406	37.8	19.45
a_1	0.0799	0.0787	0.0748
a_2	0.000059948	0.000059946	0.000077183
a_3	0.000000567	0.000000569	0.000000220
a_4	0.0000000012	0.0000000012	0.0000000024

Table A

Notice that the coefficients for the higher order terms are quickly approaching zero. This was expected because the plot looked linear. Obviously, a good approximation for the curve should be a linear equation(i.e. $G(t_{Local}) = a_0 + a_1 * t_{Local}$). To confirm this, let's look at how long it would take before the contribution of each of the terms of the polynomial would become measurable. Let's assume that the resolution of the timestamp mechanism is 1 millisecond. Table B shows how much time has to elapse before the contribution of a term would be big enough to be measurable.

There is quite a difference in time between the a_1 term and the a_2 term. The time to contribute 1 millisecond goes from 45 seconds to over 4 hours on clock X. Clock Y and Clock Z have the same characteristics.

Curve fitting techniques attempt to derive a polynomial that passes through each of the data points. The problem with the curve fitting technique is the derived polynomial function is not very accurate between

the data points when the number of data points gets greater then 10. For those cases where the measured data is polynomial in nature, the interpolation of the values of the function between the data points is good. Not all phenomenon is of a polynomial nature and for those cases the interpo-

	Clock X	Clock Y	Clock Z
a_0	0.0000	0.0000	0.0000
a_1	.0125	.0127	.0134
a_2	4.0843	4.0843	3.5995
a_3	12.0820	12.0678	16.5650
a_4	30.2100	30.1200	25.4070

Number of hours before the term contributes 1 Millisecond

Table B

lation is poor. The accuracy of the interpolation depends on similarity between the curve fitted polynomial and the phenomenon that is being measured.

Each of the above graphs has over 200 data points. In general, the derived polynomial, that passes through this many points, would have higher coefficients with larger values. For the higher order coefficients to be so small with these many data points means that the function is dominated by the lower order coefficients. Since there is only zero order and first order terms, the function is linear.

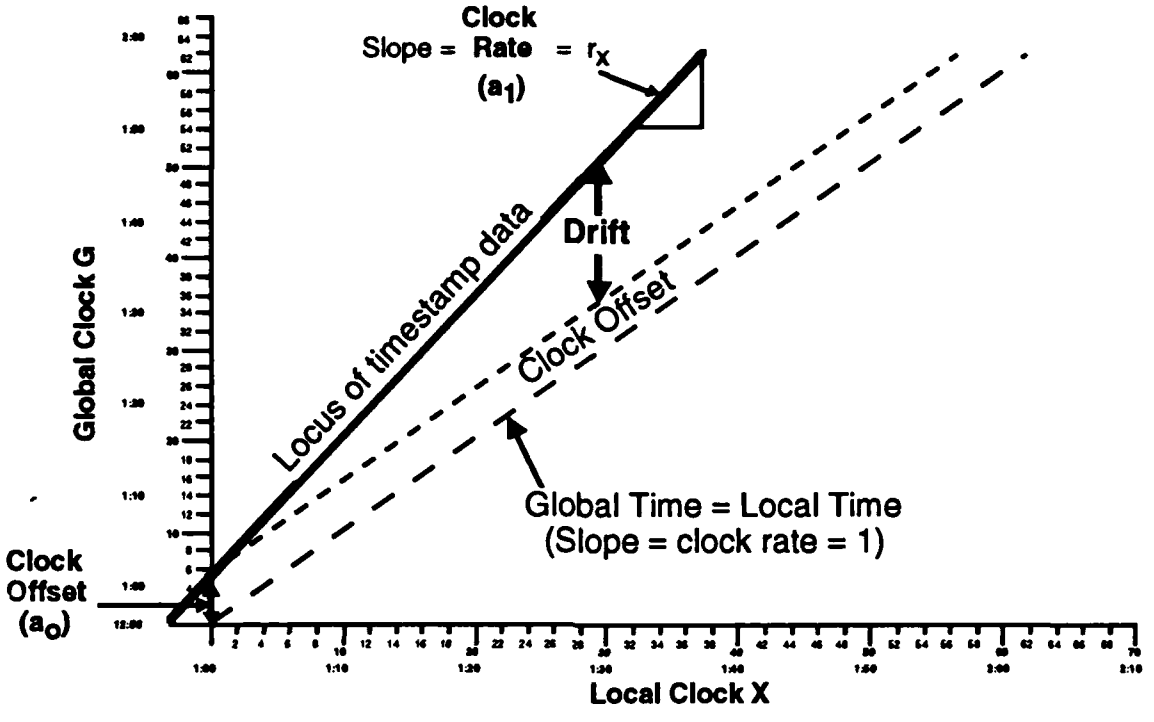
The Global Time Transformation

From Table A it is obvious that the dominant terms for all three systems are the zero and first order terms. The polynomial that best fits this locus of points is

$$G(t_{Local}) = a_0 + a_1 \cdot t_{Local} \qquad (1)$$

which is linear. The global time transformation should be a function that has the same form as Equation (1). The two coefficients of the global time

transformation, $G(t_{Local})$, are the only two values that have to be determined for the global time transformation. Figure 2D is a plot of a typical set of timestamp data. The a_0 coefficient is the clock offset which is the global time when the local time is zero. This coefficient is zero only when



The global time transformation Model
Figure 2D

the clocks are initially set to the exact time. For the global time transformation to be *correct* for all time, this term has to be zero.

The a_1 coefficient is the clock rate. The clock rate is the rate at which the clock is changing time. The terms clock rate and clock drift rate are often used interchangeably. The two terms are not the same but are related. The difference will be discussed in detail in Section 3. The a_1 coefficient in Equation (1) deals with the speed of the clock. Because it is not possible to build with today's technology two clocks that change time exactly at the same speed, the a_1 coefficient is never 1.

In the Equation (1) of the global time transformation, the * operation in the term $a_1 * t_{Local}$, has to be clarified. The coefficient a_1 is a number that is base 10 and t_{Local} is a number that is base hour-second-minute. Since the operation * is not defined for the operands of different bases, a

conversion has to be done on one of the operands. The t_{Local} operand has to be converted to a base 10 and the operation * for base 10 is used or the a_1 has to be converted to an hour-second-minute number and the operation * for hour-second-minutes be used. Since the base 10 operations are easier than hour-second-minutes operations to implement, the conversion will be to base 10.

Since the prediction of the global time transformation is base hour-second-minutes, the global time transformation must convert timestamp data (base hour-second-minute) to elapse time (base 10), then do the global transformation and then invert the transformation back to clock time(base hour-second-minute).

In summary, the global time transformation takes the clock reading from any clock in the multiclock system and calculates the corresponding clock reading for the global clock.

The only information that is needed to use the Global Time Transformation is the clock offset and the clock rate for the system's clock. A discussion of the clock rate and the clock offset is given in the Sections 3 and 4 respectively.

Section 3: Clock Rate

Definitions

This section derives the equations for calculating the clock rate. Formal definitions are given for the terms and notation that is used in the derivation of the clock rate equations. As a prelude to the derivation, background information is given on the clocking mechanisms, types of messages and measurement techniques. This information enables the reader to follow the derivation of the clock rate equations.

The Global time transformation is a polynomial function (i.e. $G(t_{local}) = a_0 + a_1 * t_{Local}$) which contains only two coefficients. In this section, the coefficient a_1, the clock rate will be discussed. The derivation of the various clock rate functions is also provided. To calculate the clock rate, data has to be measured and collected. A measurement technique has to be developed to obtain the data. This section discusses the issues associated with developing a measurement technique.

The definition of ***clock rate*** is the time rate of change of the clock. The clock rate is what defines the lengths of time for a unit of time (e.g. second). When systems have different clock rates, they have different units of time. The accumulation of the difference in the units of time is called the clock ***drift.*** Drift is part of the total difference in the clock readings. The other part is the clock offset. The clock ***offset*** is the difference in the clocks at the initial or reference time (see Figure 3A). The offset and drift both appear in the global time transformation as a_0 and a_1 in Equation (1) respectively.

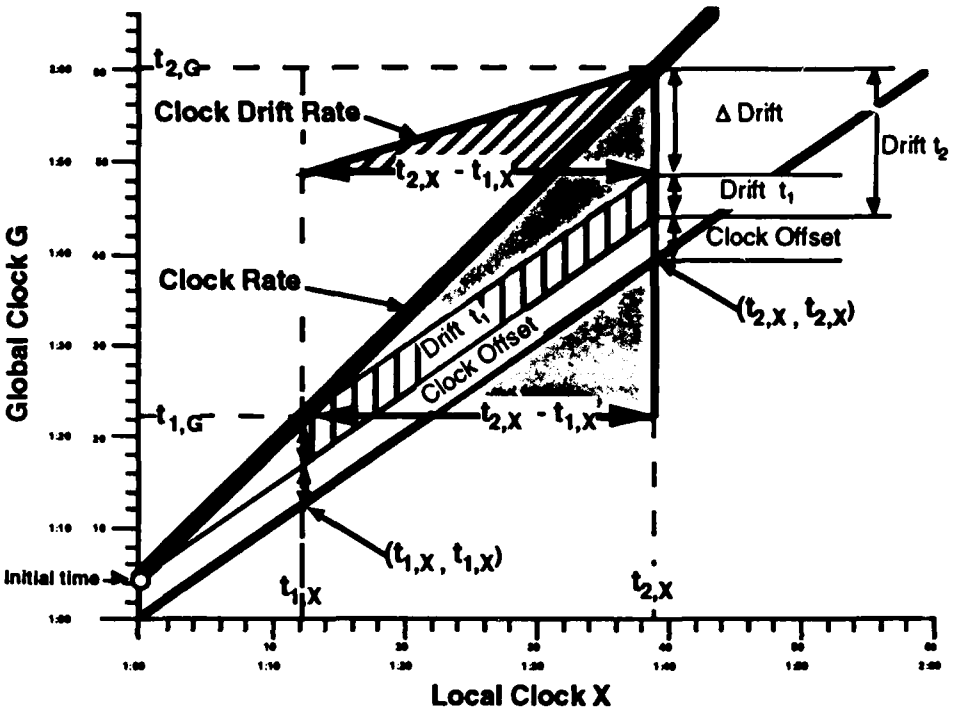
The rate that the drift is changing is called the ***clock drift rate.*** The clock rate and the drift rate are different (see Figure 3A). The clock rate is calculated using the clock readings with respect to time. The clock drift rate is calculated as the change in drift with respect to time.

The Clock Rate Equation

The clock rate for system X is the slope of the time line for the system (the thick line in figure 3A). The clock rate for system X is

$$r_x = \frac{t_{2,G} - t_{1,G}}{t_{2,X} - t_{1,X}} \qquad (2)$$

where $t_{2,G}$ is the corresponding global time for the local time $t_{2,X}$ for clock X. and $t_{1,G}$ is the corresponding global time for the local time $t_{1,X}$.



Clock rate and clock Drift rate

Figure 3A

The clock drift rate, d_x, for system X is

$$d_x = \frac{\Delta \text{ drift}}{t_{2,X} - t_{1,X}}$$

$$\begin{aligned} \Delta \text{ drift} &= \text{drift}_{t_2} - \text{drift}_{t_1} \\ &= [t_{2,G} - (t_{2,X} + \text{clock offset})] - [t_{1,G} - (t_{1,X} + \text{clock offset})] \\ &= t_{2,G} - t_{2,X} - t_{1,G} + t_{1,X} \\ &= (t_{2,G} - t_{1,G}) - (t_{2,X} - t_{1,X}) \end{aligned}$$

Therefore,

$$d_x = \frac{(t_{2,G} - t_{1,G}) - (t_{2,X} - t_{1,X})}{t_{2,X} - t_{1,X}} \quad . \qquad (3)$$

The clock rate and clock drift rate are different but are related in the following way.

$$d_X = \frac{(t_{2,G} - t_{1,G}) - (t_{2,X} - t_{1,X})}{t_{2,X} - t_{1,X}}$$

$$d_X = \frac{(t_{2,G} - t_{1,G})}{t_{2,X} - t_{1,X}} - \frac{(t_{2,X} - t_{1,X})}{t_{2,X} - t_{1,X}}$$

$$= \frac{(t_{2,G} - t_{1,G})}{t_{2,X} - t_{1,X}} - 1$$

Using Equation (2)

$$d_X = r_X - 1$$

or

$$r_X = d_X + 1 . \qquad (4)$$

For the global clock, the clock rate is $r_G = \frac{t_{2,G} - t_{1,G}}{t_{2,G} - t_{1,G}} = 1.$

Substituting this into equation (4), gives you the following

$$r_X = d_X + r_G .$$

The clock rate for clock X is the clock drift rate for clock X plus the global clock rate. Notice that if there is no clock drift rate, the local clock rate is the same as the global clock rate. When this occurs, the clock X is *accurate.*

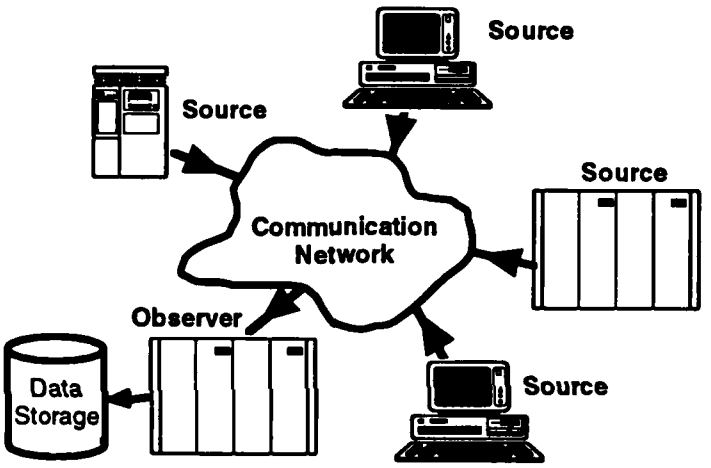
The clock rate is the rate that is used by the global time transformation as the coefficient a_1. This is because clock rate simplifies the calculations in the transformation and it is easier to use when solving the equations. To use the clock drift rate, the global time transformation would have to be modified in the first order term to

$$G(t_{Local}) = a_0 + (a_1 + 1) \cdot t_{Local} .$$

In this report, the global time transformation will use the clock rate for a_1. To calculate the clock rate, timestamp data is gathered from each of the clocks. This can be accomplished by using the observer-source model (see figure 3B). This model has at least one observer that monitors each of the sources during an experiment. The experiment is implemented in

such a way that each of the sources communicates their present clock time to the observer. The observer accepts and records the timestamp data which is then used to calculate the clock rate.

The method by which the time is communicated from the source to the observer(s) is dependent upon the clocking mechanism, the message type, and the observer-source arrangement. The clocking mechanism is the mechanism that initiates the message. The message type is the kind of message that is sent. The observer-source arrangement defines who sends messages to who.

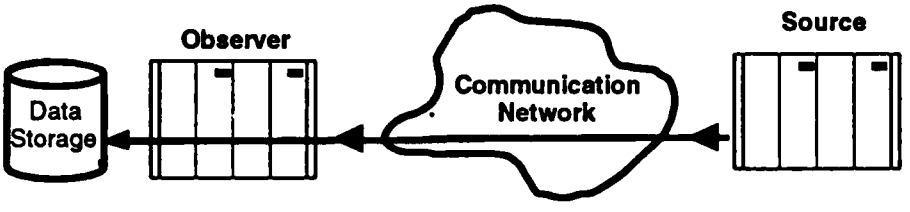


Observer-Source Model
Figure 3B

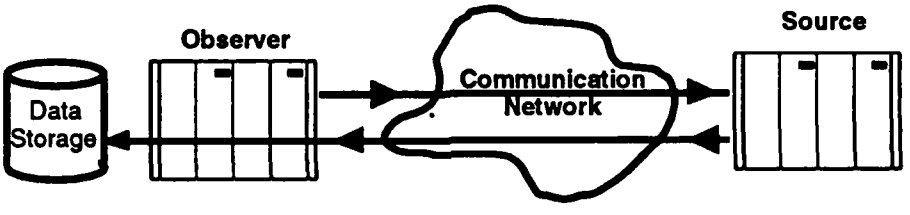
Clocking Mechanisms

The four basic types of clocking mechanisms are the chimer, the alarm, the pulser, and the repeater[Milh45]. The ***Chimer*** sends/broadcasts at *periodic intervals* the *time* in a message to the observer(s). The message contains the information of the current time. This is like a grandfather clock that chimes and the total number of chimes is the present hour. The ***alarm*** sends at a *specified time* a *signal* to the observer(s). A signal is a message that doesn't contain any information about the time. Because the time that the signal was to be sent was predetermined, the observer only needs to receive a signal from the source to know the departure time of the message. Like the name implies, an alarm is like an alarm clock that rings at a specified time. The ring doesn't tell what time it is but the listener knows because the signal

occurred at the previously specified time. The ***pulser*** sends/broadcasts at *periodic intervals* a *signal* for the observer(s). This mechanism is primarily used to synchronize multiple clocks. The ***repeater*** sends/broadcasts, *only upon request*, the *time* or *a signal* to the requester. The source acts as a "pinging" mechanism. The chimer, alarm, pulser and repeater are the basic timing mechanisms. Selecting the appropriate timing mechanism will depend on the type of message capability in the source system.



Chimer, Alarm and Pulser Model



Repeater Model

Figure 3D

Static and Dynamic Messages

The two types of messages are the static and the dynamic message types. A ***static message*** is a fixed message that was previously defined and built for the observer(s). Static messages are used primarily as signals. A ***dynamic message*** is a message that is generated every time it is requested. The dynamic message is used when the observer needs to retrieve the time from the source's clock. Some sources don't have the ability to generate dynamic messages. The type of message will depend on the experiment, the source and the clocking mechanism.

Observer-Source Arrangements

The third section of the methodology is the observer-source arrangement. This subsection identifies to whom the message is to being sent.

The three fundamental arrangements are the singular, multicast and broadcast arrangements. In the singular arrangement, the source sends the time or a signal to only one observer. In the multicast arrangement, the source sends the time or signal to a select set of observers. In the broadcast arrangement, the source indiscriminately sends the time or signal to all observers. Selecting the appropriate observer-source arrangement, the clocking mechanism and the message type help define the measurement technique to be used for the experiment.

Experiment Configurations

Using the observer-source model and the various communication methods, there are many types of experiments that can be developed to gather the data. Only two of these experiments will be discussed in this report. The first experiment involves a source that has the ability to generate dynamic messages[Mills 92]. If dynamic messages and the chimer mechanism are used, the source would extract at periodic intervals (e.g. once an hour or day) the time from the system clock put it into a message, then send the message to the observer(s). The observer(s) would then extract the time from the message and records both the time datum within the message and the arrival time of the message. If dynamic messages and the repeater mechanism are used, the observer would send a request message to the source. The source would receive the message from the observer, get the time from the system clock, put it into a message and then send it to the observer. Again, the observer would receive the message, extract the time from the message and record both the departure time and the arrival time of the message. The main advantage of the dynamic message technique is that the message contains departure time information while a static message doesn't. The main disadvantage of the dynamic message technique is that it takes time to extract the time from the clock and put into the message. This additional time distorts the timestamp data. If the time between messages is very long, the distortion has less of an impact on the calculation of the clock rate.

In the second experiment, the source only has the ability to send static messages(i.e. signals) and the clocking mechanism is the alarm. The alarm mechanism is set to a series of specified times. When the specified times arrive, the source sends a signal (i.e. static message) to the

observer(s). The observer(s), upon receiving the message, records the arrival time. The arrival time is not the departure time for the message. The arrival time is the departure time plus all the propagation delays between the systems. In the dynamic message case, the observer gets both the departure time and the arrival time. The main advantage of this experiment is that it is easy to implement. The main disadvantage is that it is hard to account for all those fluctuating delays when calculating the clock rate. There are more experiments that can be generated, but these two show how dynamic and static messages can be used to provide timestamp data to the observer .

Periodic Intervals

An issue that needs to be addressed is how to generate periodic intervals for the chimer and pulser clocking mechanisms. There are two fundamental techniques to generating periodic intervals with software. One technique is to have the software enter a wait state until the timer counts down to zero, then leaves the wait state, sends the message, resets the timer to the specified interval time and returns again to the wait state. The process is repeated over and over again during the experiment. This technique is the ***wait-state*** technique. The problem with the wait-state technique is that all that time spent outside the wait state is added to the cycle time. The actual time for one cycle is the interval time plus the additional time spent outside the wait state while the software is running in between the wait states. The additional time outside the wait state accumulates to noticeable levels within seconds. To compensate for the accumulated effects of the software processing time to the interval, the interval time for the wait state is reduced. This adjustment to the wait-state time isn't exact because of the variations in the processing time for the message and resetting the timer.

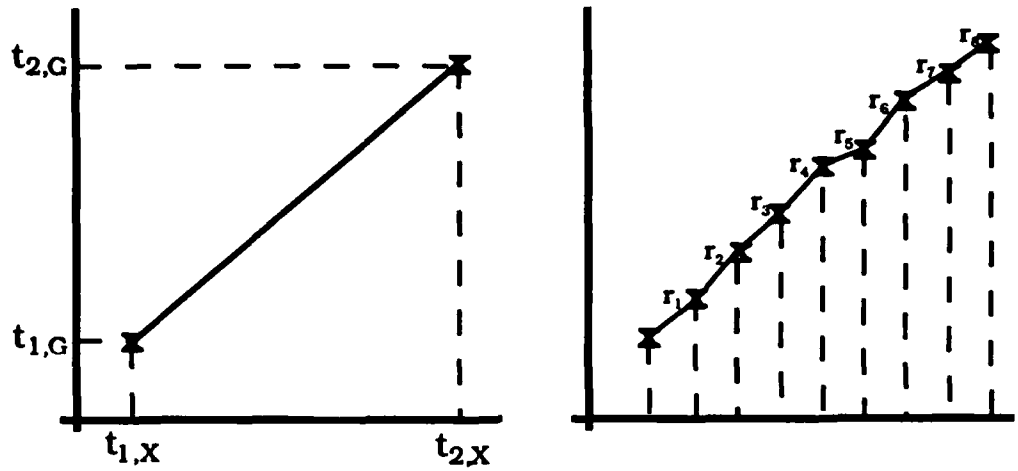
The other technique is to have the source software use the system clock and at the specified clock time(s) the software sends a message to the observer. When the message arrives, the observer records the arrival time. The time between messages is based on the clock time and not an interval of time. This technique is the ***clock-based*** technique. The problem with the clock-based technique is that the software keeps checking the clock to see if the time has expired. When the processor spends time

to check the clock, it takes the processor away from the other processing activities. This slows down the primary processing activities. To correct this problem, separate mechanism is needed to monitor the clock so that the processor is not interrupted. Because there isn't any accumulative effects with the clock-based technique, the clock-based technique is the better technique of the two techniques for generating periodic intervals.

Long Interval and Short Interval Techniques

After gathering the timestamp data, the next step is to calculate the clock rate. The two fundamental techniques for calculating the clock rate are referred to as the long interval technique and the short interval technique (see figure 3C). With the ***long interval technique***, the data (i.e., $t_{1,G}$, $t_{2,G}$, $t_{1,X}$ and $t_{2,X}$) are collected over a long period of time and then the initial measurements and the final measurements are substituted into Equation (2) to calculate the clock rate. With the ***short interval technique***, the data is collected over a shorter period of time. From the sets of interval measurements, a *set of interval clock rates* are calculated. From the set of interval clock rates, the *effective* clock rate is calculated for the system. Of the two techniques, the simplest and most precise technique is the long interval technique because the variations in data have less of an impact on the calculation. Because the size of the variations, (fractions of a millisecond), is relatively small compared to the measurement interval, (weeks or months), the results of calculations are more accurate. One problem with using the long interval technique is that it is very hard to get a system to stay on long enough for a complete set of measurements, to be taken. Usually there is a problem that interrupts the system. For example, the system is rebooted because of a software problem or the system power is turned off either manually or by a power outage. The short interval technique is used when the window of time in which you have to measure the clock rate is relatively small (e.g. a couple of days). The problem with this technique is that all the variations in the delays that occur during the measurements have to be accounted for in the calculation of the clock rate. The types of delays are numerous and hard to measure. Some of the delays are in the message firing times, the communication network, the system loading, encoding/decoding the

network protocols and the measurement process. To make the problem worse, the delays vary in time and therefore are hard to determine.



Long Interval Technique Short Interval Technique

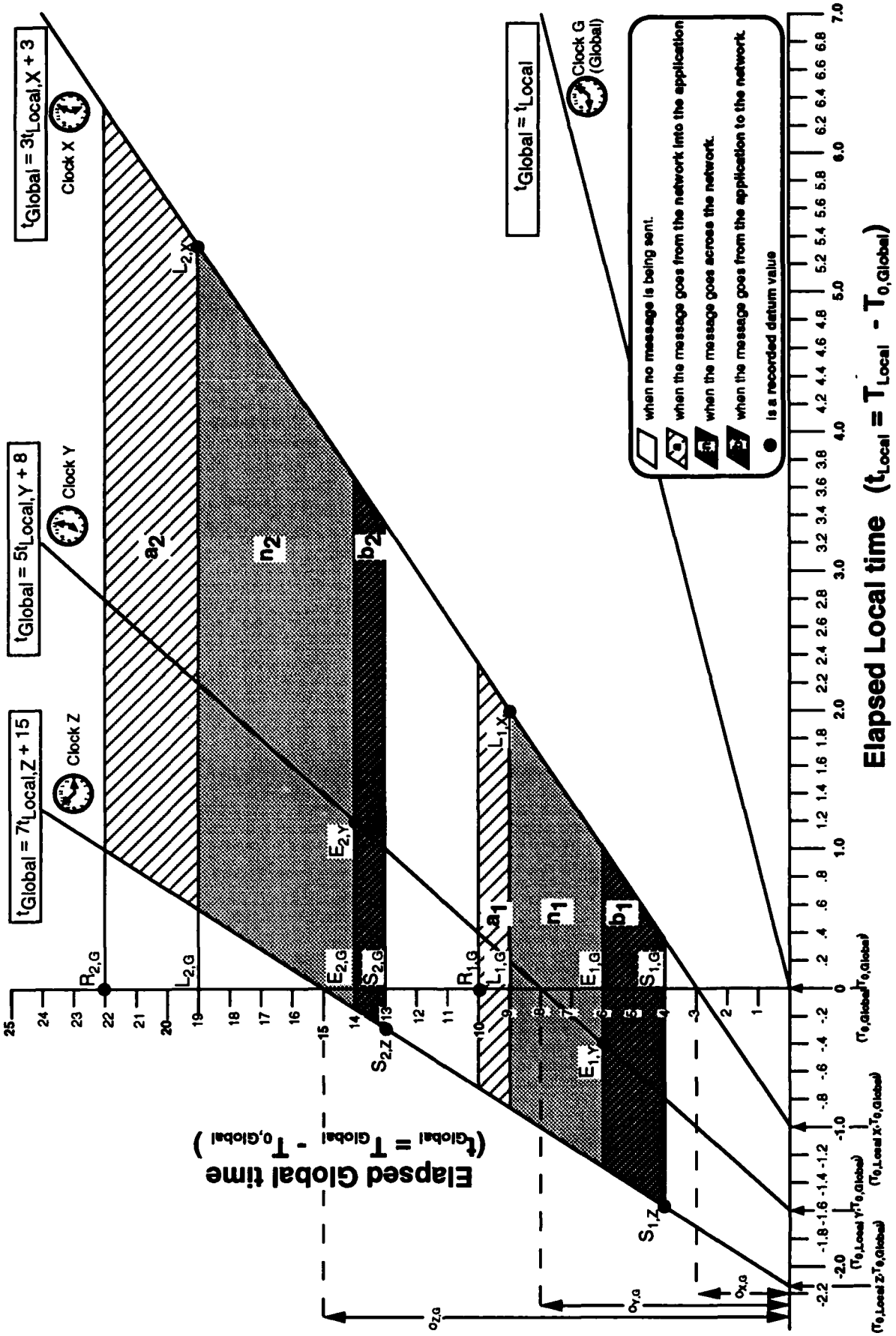
Long interval and Short interval Calculations

Figure 3C

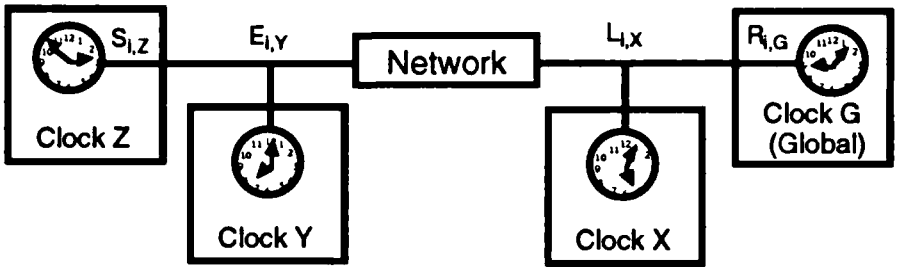
In the long interval technique, the steps are simply to take the initial measurements, take the final measurements and put them into equation (2) (see Figure 3C). The short interval technique is not quite as simple (see Figure 3C). Many more measurements required because the calculations are sensitive to measurement errors and variations in the delays. To make the calculations more accurate, the equation (s) for the clock rate must be expanded to take into account the variations in the measurement process (e.g. network effects and system loading).

A Four Clock System

The clock rate calculations are dependent on the number of delays that are encountered during the message passing process. The number of delays depend mainly on the amount network hardware there is between the message source and the observer. To derive the equations for the clock rate, a four clock system model is used. Figure 3D shows the interconnectivity of the system model. Clocks Y and Z are on the opposite side of the network than the global clock G. Clock X is on the same side of the network as the global clock G. Clock Y and X are a static message devices and clock Z is a dynamic message device.



Time plot for a Four Clock System
Figure 3E



System interconnectivity
Figure 3D

Figure 3E is used to help track the movement of the two messages as they go from source to observer. The flow of two messages from system Z to system G is depicted by the two groups of shaded areas. In both the upper and lower shaded areas, area **a** signifies the time that the message takes to go from the network to the application, area **n** signifies the time that the message is in the network and area **b** signifies the time that it takes to go from the application to the network. The elapsed global time of each event is read off of the vertical axis. The elapsed local time for an event is read off of the horizontal axis.

Let's identify the recorded data for the first message, which is the lower shaded area. Message one is Sent from system Z at time $S_{1,Z}$. According to system Y, the message Enters the network at time $E_{1,Y}$. According to the clock in system X, the message Leaves the network at time $L_{1,X}$. The message is Received by system G at time $R_{1,G}$. Four separate time lines are drawn for the four separate clocks. The time lines for clocks X, Y and Z represent measured drift data and are arbitrarily selected for the derivation of the clock rate. Figure 3E will be used in the derivation of the clock rate for the three systems. The numeric entries on the plots are not used in the derivation but can aid in following the derivation.

The notation that is used in the plot and in the derivations of the clock rates is as follows:

T is the timestamp datum or reading from the system clock.

t is the elapsed time.

• is a recorded timestamp datum.

$a_{i,A}$ is the length of time it takes for the i^{th} message to go *from the network to the application* with respect to clock A.

$b_{i,A}$ is the length of time it takes for the i^{th} message to go *from the application to the network* with respect to clock A.

$E_{i,A}$ is the elapsed time with respect to clock A when the i^{th} message **E***nters* the network.

$L_{i,A}$ is the elapsed time with respect to clock A when the i^{th} message **L***eaves* the network.

$n_{i,A}$ is the length of time (**n***etwork propagation delay*) it takes for the i^{th} message with respect to clock A.

$o_{A,G}$ is the **O***ffset* of clock A with respect to clock G.

$r_{A,B}$ is the **r***ate* that clock A is drifting from clock B.

$S_{i,A}$ is the elapsed time with respect to clock A of when the i^{th} message was **S***ent*.

$R_{i,A}$ is the elapsed time with respect to clock A of when the i^{th} message was **R***eceived*.

In Figure 3E, there are two types of variables, those with recorded data and those without recorded data e.g., $E_{2,Y}$ and $E_{1,G}$ respectively. The • for $E_{2,Y}$ signifies that it is a recorded value. $E_{1,G}$ is not a recorded datum because it doesn't have a •.

Derivation of the Clock Rate Equations

There are only three clock rates that have to be calculated because one of the clocks is the reference or global clock. The three clock rates are r_X, r_Y, and r_Z. By definition,

$$r_x = \frac{L_{2,G} - L_{1,G}}{L_{2,X} - L_{1,X}} \quad . \tag{5}$$

Since $L_{1G} = R_{1,G} - a_{1,G}$ and $L_{2,G} = R_{2,G} - a_{2,G}$ where $a_{1,G}$ is the length of time it takes the i^{th} message to propagate from the network to the application as measured by clock G

$$\begin{aligned} L_{2,G} - L_{1,G} &= (R_{2,G} - a_{2,G}) - (R_{1,G} - a_{1,G}) \\ &= (R_{2,G} - R_{1,G}) - (a_{2,G} - a_{1,G}) \end{aligned}$$

Substituting this into Equation (5)

$$r_X = \frac{(R_{2,G} - R_{1,G}) - (a_{2,G} - a_{1,G})}{L_{2,X} - L_{1,X}}$$

$$= \frac{(R_{2,G} - R_{1,G})}{L_{2,X} - L_{1,X}} - \frac{(a_{2,G} - a_{1,G})}{L_{2,X} - L_{1,X}} \qquad (6)$$

Note: From the data available, it is not possible to *directly* calculate $a_{2,G}$-$a_{1,G}$.

$$\begin{aligned} a_{2,G} - a_{1,G} &= L_{2,G} - L_{1,G} \\ &= (R_{2,G} - L_{2,G}) - (R_{1,G} - L_{1,G}) \\ &= (R_{2,G} - R_{1,G}) - (L_{2,G} - L_{1,G}) \end{aligned}$$

Since $L_{1,G} = r_X * L_{1,X} + O_{X,G}$ and $L_{2,G} = r_X * L_{2,X} + O_{X,G}$ then to calculate $L_{2,G} - L_{1,G}$ the value for r_x must be approximated because it is not known. The approximation for r_X is used to calculate $L_{2,G} - L_{1,G}$ which is then used to calculate $a_{2,G} - a_{1,G}$. After substituting $a_{2,G} - a_{1,G}$ into Equation (6), the resulting calculation is equal to the value that was selected for the approximation. If $R_{2,G} - R_{1,G} >> a_{2,G} - a_{1,G}$ then

$$r_X \approx \frac{R_{2,G} - R_{1,G}}{L_{2,X} - L_{1,X}} \quad . \qquad (7)$$

This is a good assumption because the size of $a_{i,G}$ is relatively smaller than the network delay and therefore $a_{i+1,G} - a_{1,G}$ is very small, since $a_{i+1,G}$ and $a_{i,G}$ both are > 0.

The next clock rate to drive is r_Y. The calculation of the clock rate for system Y starts with the definition of clock rate,

$$r_Y = \frac{E_{2,G} - E_{1,G}}{E_{2,Y} - E_{1,Y}} \quad . \tag{8}$$

Since $E_{2,G} = R_{2,G} - a_{2,G} - n_{2,G}$ and $E_{1,G} = R_{1,G} - a_{1,G} - n_{1,G}$ then

$$\begin{aligned} E_{2,G} - E_{1,G} &= (R_{2,G} - a_{2,G} - n_{2,G}) - (R_{1,G} - a_{1,G} - n_{1,G}) \\ &= (R_{2,G} - R_{1,G}) - (a_{2,G} - a_{1,G}) - (n_{2,G} - n_{1,G}) \quad . \end{aligned} \tag{9}$$

Substituting this into Equation (8)

$$r_Y = \frac{(R_{2,G} - R_{1,G}) - (a_{2,G} - a_{1,G}) - (n_{2,G} - n_{1,G})}{E_{2,Y} - E_{1,Y}}$$

$$= \frac{(R_{2,G} - R_{1,G})}{E_{2,Y} - E_{1,Y}} - \frac{(a_{2,G} - a_{1,G})}{E_{2,Y} - E_{1,Y}} - \frac{(n_{2,G} - n_{1,G})}{E_{2,Y} - E_{1,Y}} \quad . \tag{10}$$

Since $a_{2,G}$-$a_{1,G}$ is relatively small, the Equation (9) can be approximated by

$$r_Y \approx \frac{(R_{2,G} - R_{1,G})}{E_{2,Y} - E_{1,Y}} - \frac{(n_{2,G} - n_{1,G})}{E_{2,Y} - E_{1,Y}} \quad . \tag{11}$$

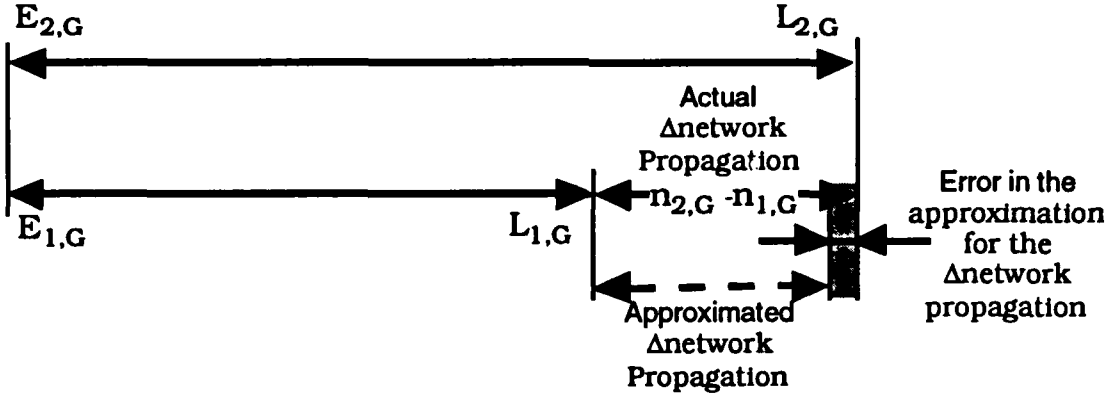
The only unknown variable in this equation is $n_{2,G}$-$n_{1,G}$. Since $n_{2,G} = L_{2,G} - E_{2,G}$ and $n_{2,G} = L_{2,G} - E_{2,G}$ then

$$\begin{aligned} n_{2,G} - n_{1,G} &= (L_{2,G} - E_{2,G}) - (L_{1,G} - E_{1,G}) \\ &= (L_{2,G} - L_{1,G}) - (E_{2,G} - E_{1,G}) \end{aligned} \tag{12}$$

Using the definitions for r_X and r_Y, $L_{2,G} - L_{1,G} = r_X*(L_{2,X} - L_{1,X})$ and $E_{2,G} - E_{1,G} = r_Y*(E_{2,Y} - E_{1,Y})$. Substituting these into Equation (12)

$$n_{2,G} - n_{1,G} = r_X*(L_{2,X} - L_{1,X}) - r_Y*(E_{2,Y} - E_{1,Y}) \quad . \tag{13}$$

The term ($n_{2,G}$ - $n_{1,G}$) is the change in the network propagation times between messages 2 and 1. The purpose of this term in Equation (10) is to eliminate the effects of the variation in the network propagation. Since ($n_{2,G}$ - $n_{1,G}$) is dependent on r_X and r_Y which are unknown, approximate values for r_X and r_Y are used (i.e. r'_X and r'_Y respectively), see Figure 3F. The better the approximation for r'_X and r'_Y the better the resulting calculation for r_X and r_Y. Which value to use for the approximation is up to the analyst. A couple of candidate approximations for the clock rates are the last calculated clock rate or the clock rate derived from the initial and final measured values. Since the error in the approximation to the Δnetwork propagation is smaller then the Δnetwork propagation, the adjustment with the approximated clock rate reduces the impact of the variations in network effects. This will provide a much better value for the clock rate, see Figure 3F. Since ($E_{2,Y}$ - $E_{1,Y}$) is a factor of the term r_Y*($E_{2,Y}$ - $E_{1,Y}$), the linear matrix equation solution to the problem is not feasible. The ($E_{2,Y}$ - E_{1Y}) factor



ΔNetwork propagation error

Figure 3F

would eliminate the r_Y variable (see Equation (15)). The approximation for ($n_{2,G}$ - $n_{1,G}$) is

$$n_{2,G} - n_{1,G} \approx r'_X * (L_{2,X} - L_{1,X}) - r'_Y * (E_{2,Y} - E_{1,Y}) . \qquad (14)$$

Substituting this into Equation (12)

$$r_Y = \frac{(R_{2,G} - R_{1,G})}{E_{2,Y} - E_{1,Y}} - \frac{\left(r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y})\right)}{E_{2,Y} - E_{1,Y}}$$

$$\mathbf{r_Y = \frac{(R_{2,G} - R_{1,G}) - r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y})}{E_{2,Y} - E_{1,Y}}} \quad (15)$$

The next clock rate to drive is r_Z. The calculation of the clock rate for system Z starts with the definition of clock rate,

$$r_Z = \frac{S_{2,G} - S_{1,G}}{S_{2,Z} - S_{1,Z}} \quad . \quad (16)$$

Since $S_{2,G} = E_{2,G} - b_{2,G}$ and $S_{1,G} = E_{1,G} - b_{1,G}$ then

$$\begin{aligned} S_{2,G} - S_{1,G} &= (E_{2,G} - b_{2,G}) - (E_{1,G} - b_{1,G}) \\ &= (E_{2,G} - E_{1,G}) - (b_{2,G} - b_{1,G}) \quad . \end{aligned}$$

Substituting this into Equation (16)

$$\begin{aligned} r_Z &= \frac{(E_{2,G} - E_{1,G}) - (b_{2,G} - b_{1,G})}{S_{2,Z} - S_{1,Z}} \\ &= \frac{(E_{2,G} - E_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(b_{2,G} - b_{1,G})}{S_{2,Z} - S_{1,Z}} \quad . \end{aligned} \quad (17)$$

Substituting Equation (9) for $E_{2,G} - E_{1,G}$

$$r_Z = \frac{(R_{2,G} - R_{1,G}) - (a_{2,G} - a_{1,G}) - (n_{2,G} - n_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(b_{2,G} - b_{1,G})}{S_{2,Z} - S_{1,Z}}$$

$$= \frac{(R_{2,G} - R_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(a_{2,G} - a_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(n_{2,G} - n_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(b_{2,G} - b_{1,G})}{S_{2,Z} - S_{1,Z}} \quad .$$

$$= \left[\frac{(R_{2,G} - R_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(n_{2,G} - n_{1,G})}{S_{2,Z} - S_{1,Z}}\right] - \left[\frac{(a_{2,G} - a_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(b_{2,G} - b_{1,G})}{S_{2,Z} - S_{1,Z}}\right] \quad (18)$$

Like $a_{2,G} - a_{1,G}$, $b_{2,G} - b_{1,G}$ is relatively small. The only difference between $a_{2,G} - a_{1,G}$ and $b_{2,G} - b_{1,G}$ is that $a_{2,G} - a_{1,G}$ is the difference in the length of time it takes for a message to go from the network to the application and $b_{2,G} - b_{1,G}$ is the difference in the length of time it takes for a message to go from the application to the network. Since $a_{2,G} - a_{1,G}$ and $b_{2,G} - b_{1,G}$ are relatively small, the Equation (18) can be approximated by

$$r_Z = \frac{(R_{2,G} - R_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(n_{2,G} - n_{1,G})}{S_{2,Z} - S_{1,Z}} . \quad (19)$$

The only unknown in this equation is $n_{2,G} - n_{1,G}$. Using Equation (14) into Equation (19) to get

$$r_Z = \frac{(R_{2,G} - R_{1,G})}{S_{2,Z} - S_{1,Z}} - \frac{(r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y}))}{S_{2,Z} - S_{1,Z}}$$

$$r_Z = \frac{(R_{2,G} - R_{1,G}) - r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y})}{S_{2,Z} - S_{1,Z}} . \quad (20)$$

The Clock Rate Equations

The clock rate equations derived in this section are the equations that are needed to calculate the a_1 coefficients of the global time transformation. These equations are

$$r_X \approx \frac{(R_{2,G} - R_{1,G})}{L_{2,X} - L_{1,X}} \quad \text{for } R_{2,G} - R_{1,G} \gg a_{2,G} - a_{1,G} \quad (21)$$

$$r_Y = \frac{(R_{2,G} - R_{1,G}) - r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y})}{E_{2,Y} - E_{1,Y}} \quad (22)$$

$$r_Z = \frac{(R_{2,G} - R_{1,G}) - r'_X \cdot (L_{2,X} - L_{1,X}) - r'_Y \cdot (E_{2,Y} - E_{1,Y})}{S_{2,Z} - S_{1,Z}} . \quad (23)$$

Equation (21) is used when the data comes from a monitoring system on the same side of the network as the global clock. Equation (22) is use when the data comes from a monitoring system on the other side of the network from the global clock. Equation (23) is used by a computer system on the other side of the network. In the next section, the other coefficient for the global time transformation, a_0 is discussed.

Section 4: Clock Offset

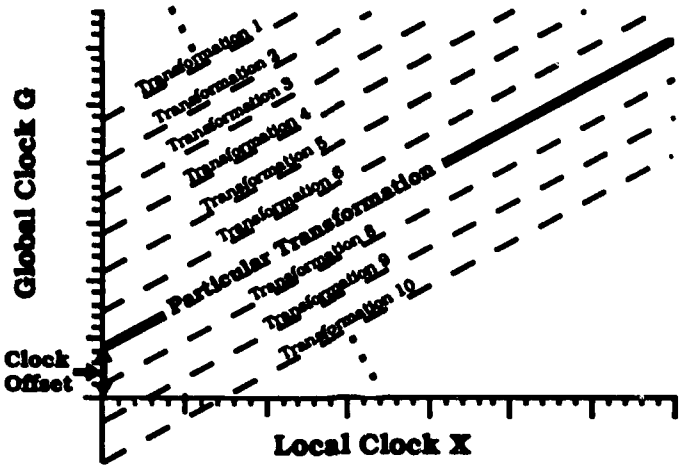
Definitions

This section describes a technique for measuring and calculating the clock offset. Since the clock offset is sensitive to delays, a detail breakdown of the delays in the message path is given for both the static and dynamic message techniques. The later part of the section discusses the tradeoffs between the static and dynamic message technique, which will help the reader select the proper technique for measuring the clock offset.

In the Global Time Transformation, there are two coefficients. The coefficient a_1 is the clock rate which was discussed in Section 3. The other coefficient a_0 is the clock offset and is the topic of this section. The definition of ***clock offset*** is the difference in time from the local clock reading to the global clock reading at the initial or reference time (see Figure 4A).

Once the clock offset is measured, it can be used with the clock rate to form the global time transformation. The accuracy of the transformation depends on the accuracy of the clock offset measurement. Measuring the clock offset to be *correct* to within a specified error (e.g. 1 millisecond) is more difficult to do than the clock rate. The difficulty is that the clock offset measurements are taken within a few seconds, unlike the clock rate which is hours or days.

The clock rate measurements are usually taken at the beginning of the experiment. Since the clocks are aligned at the beginning of the experiment, the offset should be measured right after the alignment. This is primarily due to the fact that the magnitude of the drift effects are smallest at that the beginning of the experiment. The clock rate defines a *set* of global time transformations for that system and the clock offset determines which one of the set is the *particular* transformation for this experiment (see Figure 4A). Since there is a different clock offset for every experiment, every experiment has a different global time transformation.



Set of time transformations for clock X

Figure 4A

The clock rate only needs to be measured once if the operational environment for the clock is relatively constant or the experiment has a short time duration. A new clock offset must be measured for each experiment or when the clock is changed. A change to the clock can occur intentionally (e.g. a time daemon or a manual system shutdown) or unintentionally (e.g. power outage due to lightning or an automatic system shutdown if the temperature gets too hot). When the data for the experiment is collected both before and after a system interruption, the analyst must use *two* global time transformations for the transformation step. One time transformation for the data collected before the interruption and another time transformation for the data collected after the interruption.

Clock Offset Technique

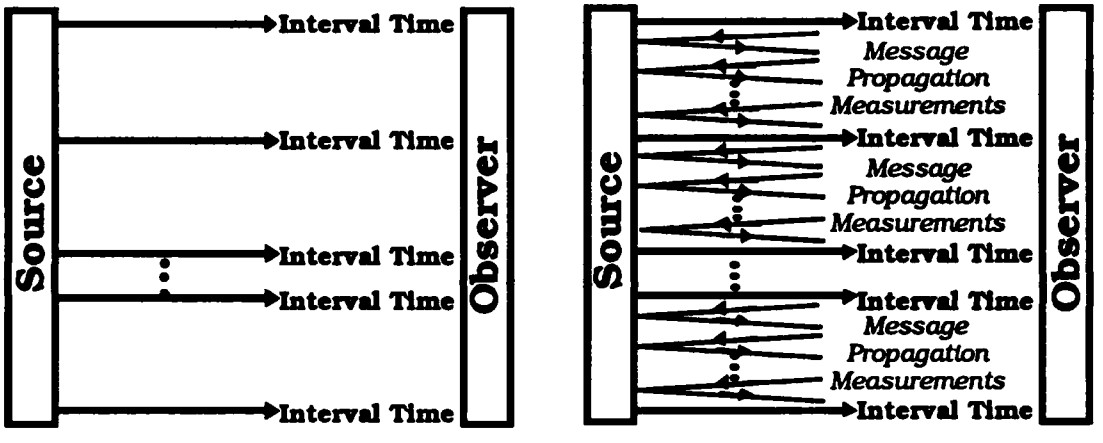
The clock offset is measured using either a static or a dynamic message technique. The static and dynamic techniques used to calculate the clock offset are different than those used for the clock rate. The static message technique used for the clock offset uses both the interval time and the message propagation measurements, unlike the clock rate which uses only the interval time measurements (see Figure 4B). The reason is that the clock rate is a function of the change in time while the clock offset is a length of time. The message propagation measurement is needed to calculate the clock offset.

Message Propagation

The reason for measuring the message propagation is that the observer can calculate the departure time of the interval message from the arrival time of the interval message if the observer knows how long it took the message to get to the observer. The calculation is

$$t_{Arrival,G} = t_{Departure,G} + \text{Message Propagation}_G$$
$$t_{Departure,G} = t_{Arrival,G} - \text{Message Propagation}_G \quad (21)$$

Where $t_{Departure,G}$ and $t_{Arrival,G}$ are the message departure and arrival times respectively, in *global* time units, and the Message Propagation



Clock Rate Technique

(Interval Times Only)

Clock Offset Technique
(Interval Times and Message Propagation Measurements)

Comparison of the Clock Rate and Clock Offset Techniques
Figure 4B

is the length of time, in global clock units, that the message takes to go from the source to the observer. Note: The calculated departure time is in global clock time units which is different than the departure time in clock X time units. This difference is noticeable when the dynamic message technique is used because the message contains the departure time in *clock X* time units. The departure times extracted from the dynamic message will be different from the calculated departure time because the clock time units are different. The next step is to measure the message propagation delay.

To measure the message propagation, a test message is sent from the observer to the source and the source returns the message to the observer (i.e. the observer "pings" the source) and the round trip time is measured. The simplest method for calculating the message propagation is to divide the round trip time by two. Since the propagation in one direction is not equal to the propagation in the other direction, the approximation for the message propagation is very rough. The approximation is rough because the propagation delays vary.

To get a more precise value for the message propagation, a set of measurements are taken and used to calculate the message propagation. Between the interval time measurements (see Figure 4B), another set of measurements are collected which are the trip times for messages going from the observer to the source and return. The number of round trip time measurements varies quite a bit depending on the distance between the source and the observer, number of time intervals and the width of the time intervals. Since most systems are stationary, the distance between the source and observer is fixed. Therefore, only the width of the intervals and the number of intervals are under your control. The length of time between the interval times must be wide enough that an adequate number of measurements can be taken (see Figure 4B). The number of measurements must be large enough to show the true variability of the message propagation. Larger time intervals allow for more message propagation measurements. In our experiments, the observer and server were relatively close, so our set of measurements was defined to be a collection of ten intervals with a width of one second and averaged approximately 10 round trip times (i.e. pings) per interval. The total number of propagation measurements were 100 for the experiment.

Once the measurements have been taken, the analyst has multiple methods for approximating the message propagation. One method is to observe the message propagation times that where taken before and after the interval time measurement. Select those intervals where the message propagation measurements before and after the interval time measurement are relatively constant. If the times are constant or nearly constant, then you can assume that dividing the propagation by two is a good approximation to the message propagation. The hidden assumption is that the message propagation in one direction is the same as the other direc-

tion. This is not the case even though the round trip times around a time interval measurement are constant.[3] In general, *message propagation time is direction dependent.* If the analyst assumes that the propagation is the same in both directions, the approximation will be in error. The amount of the error in the approximation is dependent on the relative size of the difference in the propagation times and the length of the message propagation. The multiple message measurements give the analyst a means for developing confidence in the propagation approximation.

To get an even better approximation for the message propagation, statistical methods can be used. Since the in-house effort had cost and manpower limits, work in this area was not performed. For that reason, the statistical analysis method will not be discussed in this report.

Static Message Technique

Once the message propagation has been determined, the next step is to calculate the clock offset. By definition the clock offset is

$$\text{Clock Offset} = t_{i,G} - t_{i,X}$$

where $t_{i,G}$ is the time event i occurred according to clock G and $t_{i,X}$ is the time event i occurred according to clock X. If the event i is the departure of an interval time message from the source, then $t_{i,G}$ is $t_{Departure,G}$ (i.e. the departure time of the interval time message according to clock G), and $t_{i,X}$ is $t_{Departure,X}$ (i.e. the departure time of the interval time message according to clock X). The clock offset in terms of the departure times of the interval time message is

$$\text{Clock Offset} = t_{Departure,G} - t_{Departure,X} \cdot$$

[3] If the message propagation is constant then the round trip times will be constant. The inverse is not true.

Substituting Equation (21) into this equation

$$\text{Clock Offset} = (t_{\text{Arrival,G}} - \text{Message Propagation}_{\text{G}}) - t_{\text{Departure,X}} \quad (22)$$

$t_{Arrival,G}$ *is the observed arrival time* for the message. $t_{Departure,X}$ *is the predetermined departure time* for the interval time message. For example, if the message departs at 1:00 PM and arrivals at 1:06 PM and the message propagation is a minute, then the clock offset is 5 minutes.

To make the calculation of the message propagation even more accurate, a refinement can be made to the departure time. The refinement compensates for the fact that the source does not give an instantaneous response to the ping message. There is a delay within the source. The delay is broken up into three parts. The first part is the length of time that it takes to sense the ping message. The second part is the time that it takes to determine how to respond to the ping message. The third part is executing the response to the ping message. The total time for all three parts can be significant enough to impact the precision of the clock offset calculation. Figure 4C shows the response time of a LAN Protocol Analyzer that was a source during the clock offset measurements using static message technique.

The static message technique is a good technique for calculating the clock offset because additional measurements can be taken to refine the calculation of the current message propagation time. Furthermore, the static message technique provides many opportunities to improve the precision of the clock offset calculation (e.g. statistical methods).

Dynamic Message Technique

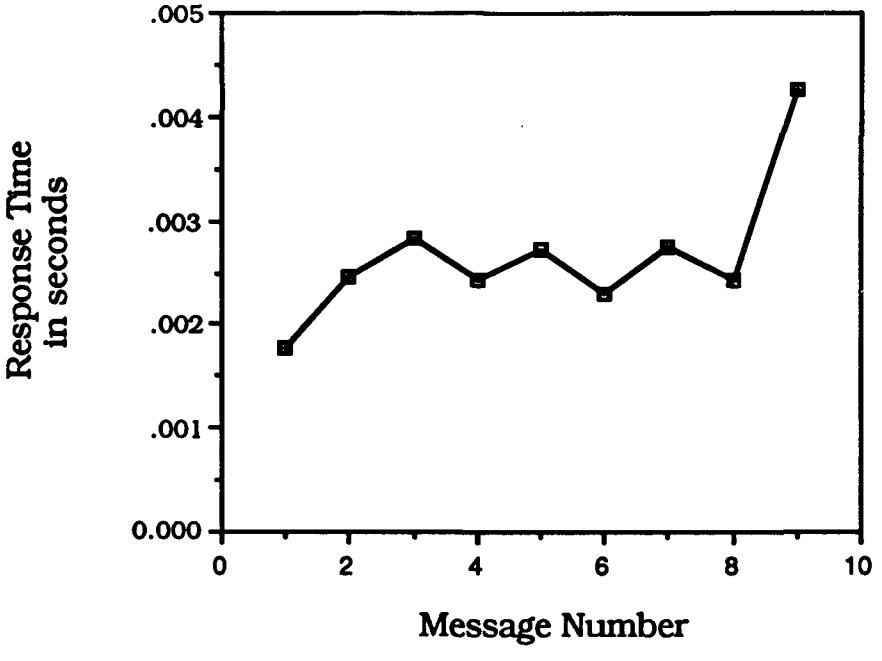
The dynamic message technique is much simpler but is not as precise as the static message technique. In the dynamic message technique (see Figure 4D), the observer gets the present time from the observer's clock and puts it into a message(1). The message is then sent out over the network to the source. The source receives the message, gets the arrival time of the message for the clock and puts it into the incoming message with the departure time(2). Since it takes time to process the message at the source, the source gets a new departure time and puts it into the message with the other departure time and arrival time(3). The message

is then sent out over the network to the observer. The observer receives the return message, gets from the clock the arrival time of the returning message and puts it into the message and stores it away(4).

The stored departure and arrival times are used to calculate the clock offset [Mills 92] as follows:

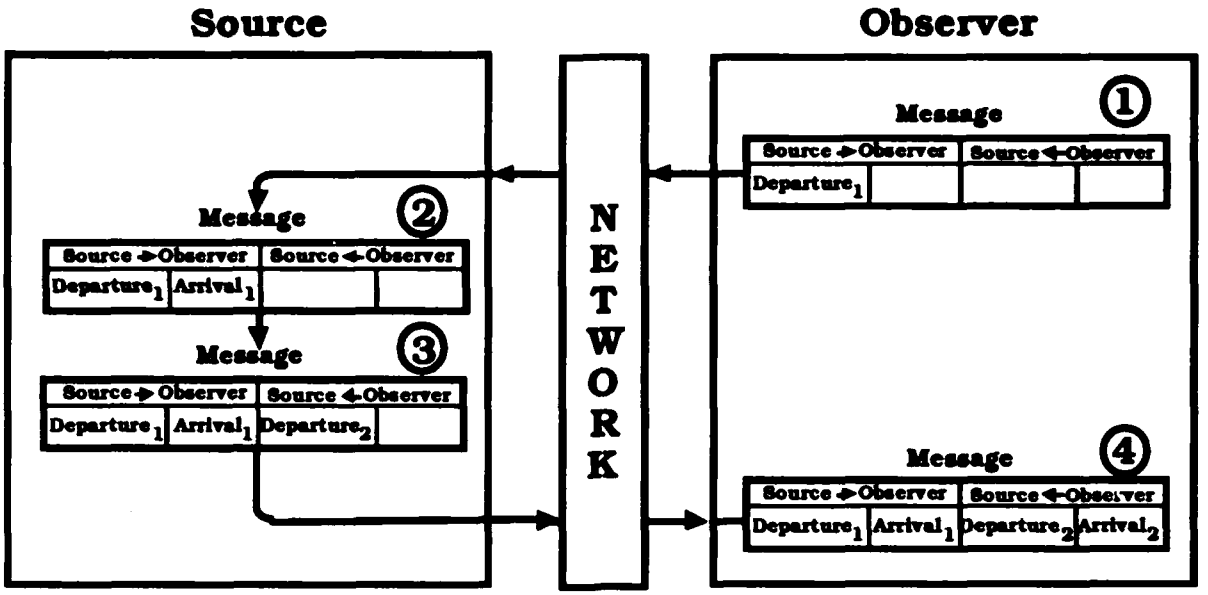
$$\text{Clock Offset} = \frac{\left(t_{Arrival,2} - t_{Departure,2}\right) + \left(t_{Arrival,1} - t_{Departure,1}\right)}{2} \quad .(23)$$

This result is a rough approximation to the clock offset because it assumes that the propagation in both directions are the same. The calculated value has just about the same accuracy as the static message technique that uses the divide-by-two approximation for the message propagation.



LAN Protocol Analyzer Response
Figure 4C

Because this technique takes into consideration the response time of the source to the ping message, it is more precise then the divide-by-two static message technique.

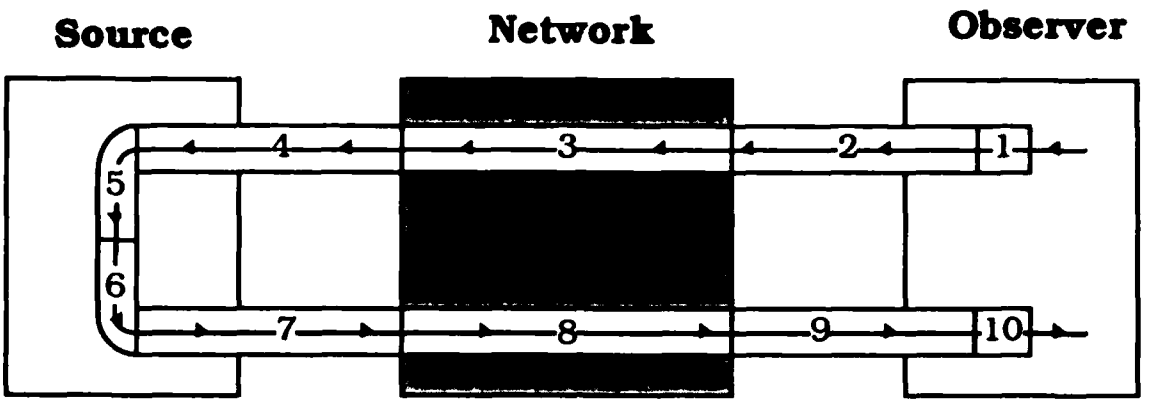


The Dynamic Message Technique for the clock offset
Figure 4D

There is very little that can be done to improve the dynamic message technique. Attempts to refine the calculation usually result in distorting the data and introducing more unknowns than there were before the refinement. The problem gets complicated very quickly.

To understand what makes the approximation a rough approximation, a closer look at the message path is needed. In Figure 4E, the message path is broken up into its components, see Figure 4E. Starting at the observer(1), the time is obtained from the clock and put into the message. The message then goes from the application in the observer down through the protocol layers(2) across the network(3) up through the protocol layers at the source to the application(4) where the time is obtained from the clock on the source clock and put into the message(5). The process is repeated but this time the message goes from the source to the observer. This breakdown has 10 steps that the message goes through and each one of them is an unpredictable delay. Using the dynamic message technique, it is hard to isolate the arrival and departure times for each of these steps without perturbing the measurements.

The inability of the dynamic message technique to account for the variation in the message propagation time makes the technique less desirable then the static message technique. The choice of the technique



The time required
1 to get and store the departure time for the message
2 for the message to propagate from the observer application to the network
3 for the message to go from the observer side to the source side of the network
4 for the message to propagate from the network to the source application program
5 to get the arrival time of the message and put it into the message
6 to get the departure time of the message and put it into the message
7 for the message to propagate from the source application program to the network
8 for the message to go from the source side to the observer side of the network
9 for the message to propagate from the network to the source application program
10 to get and store the arrival time in the message

The Path for Dynamic Messages
Figure 4E

will depend on the message generating capabilities of the system and the data resolution requirements for the analysis. If the system has only a static capability, then there is only one choice. If the system is capable of doing both the static and dynamic message techniques, then the issue is how precise does the offset have to be. If data requirements are in milliseconds or larger, then the dynamic message technique is preferred because it requires fewer measurements. If the events are in fractions of a millisecond, then the static message technique is the better choice.

The clock offset is a very important part of the global time transformation. The techniques described in this section are adequate for most types of analysis. For others, more precision is required and those situations require a more elaborate measurement scheme to improve the precision of the global time transformation. This effort didn't have enough time or money to research this issue.

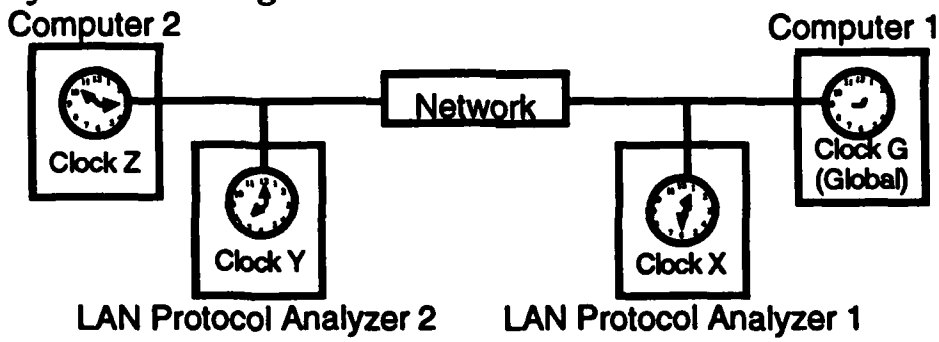
Section 5: Verification

Approach

In Sections 2, 3 and 4, the global time transformation, clock rate and the clock offset were discussed along with the measurement techniques to obtain them. In this section, those techniques were applied to a real system and a comparison is made between the global time transformation predictions and the measured clock values.

To do the comparison, two sets of measurements have to be made. The first set is used to determine the global time transformation. The second set is used to measure clock drift (i.e. via the clock offset) over a time interval. The clock rate from the first set of measurements and the first clock offset from the second set of measurements determine the *particular* global time transformation for this experiment. The global time transformation predictions of the clock offset were made over the same time interval. These predictions and the second set of measurements were compared.

Both sets of measurements used the same configuration (see Figure 5A). The configuration is a simple two node network with a computer at both ends of the network, like the four clock system discussed earlier. The two parallel computers are about 50 miles apart, and are capable of doing dynamic messages. To monitor the flow of the messages from one computer to the other computer, a LAN Protocol Analyzer (LANPA) is placed at each end of the network. These LANPAs are not capable of generating dynamic messages.

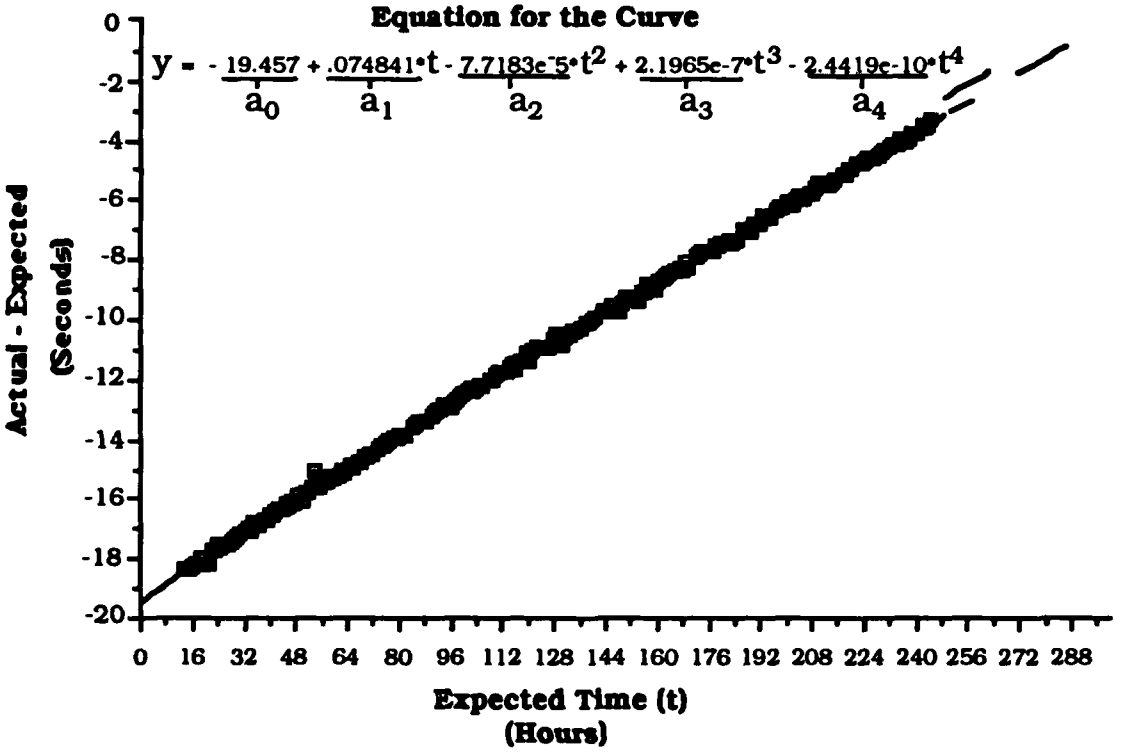


Verification Configuration
Figure 5A

Clock Rate Calculation

The first set of measurements were to determine the clock rate. The three clock rates were calculated from three separate groups of measurements. Each group of measurements contained a collection of five experiments that were run on one of the systems. Each experiment was run as long as possible and was terminated whenever there was an interruption in the measurement process. The five experiments that made up the group ran for 45, 68, 113, 143, and 233 hours.

The function of Computer 1 was to be the observer, the controller for the measurement process and the reference clock (i.e. the global clock) for the experiment. The clock rate measurements were taken of LANPA 1, LANPA 2 and Computer 2. Because the LANPAs were not capable of generating dynamic message, the static message technique was used for the LANPA measurements. Even though, Computer 2 was capable of doing dynamic messages, the static message techniques were used so that all the clock rates would be derived from the same static message technique.



A plot of the 233 hour experiment for Computer 2

Figure 5B

One of the plots for Computer 2, (the 233 hours experiment), is shown in Figure 5B. The little boxes in the plot represent a single measurement.
The line that goes through all the boxes is the linear curve fit to the data. The equation for the curve is also given in the plot. The rest of the experiments for Computer 2 and the experiments for the two LANPAs have similar linear characteristics and are not shown.

The next step is to determine the clock rate for the three systems. The clock rate can be calculated by four different methods. Two of the methods used curve fitting techniques. One method uses a simple calculation to approximate the clock rate. The last method uses the equations that were derived in Section 3.

In each of the curve fitting methods, the curve fitting technique has to be applied twice to calculate the clock rate. The two curve fitting techniques used were the least squares fitting and the polynomial fitting techniques. One of the curve fitting methods uses the polynomial fitting technique first and the least squares fitting technique second in the calculation of the clock rate. The other curve fitting method used the least squares fit for both calculations. The only difference between the two methods for calculating the clock rate is in the first step where either the polynomial fit or the least squares fit techniques are applied. Only the polynomial fitting technique will be discussed.

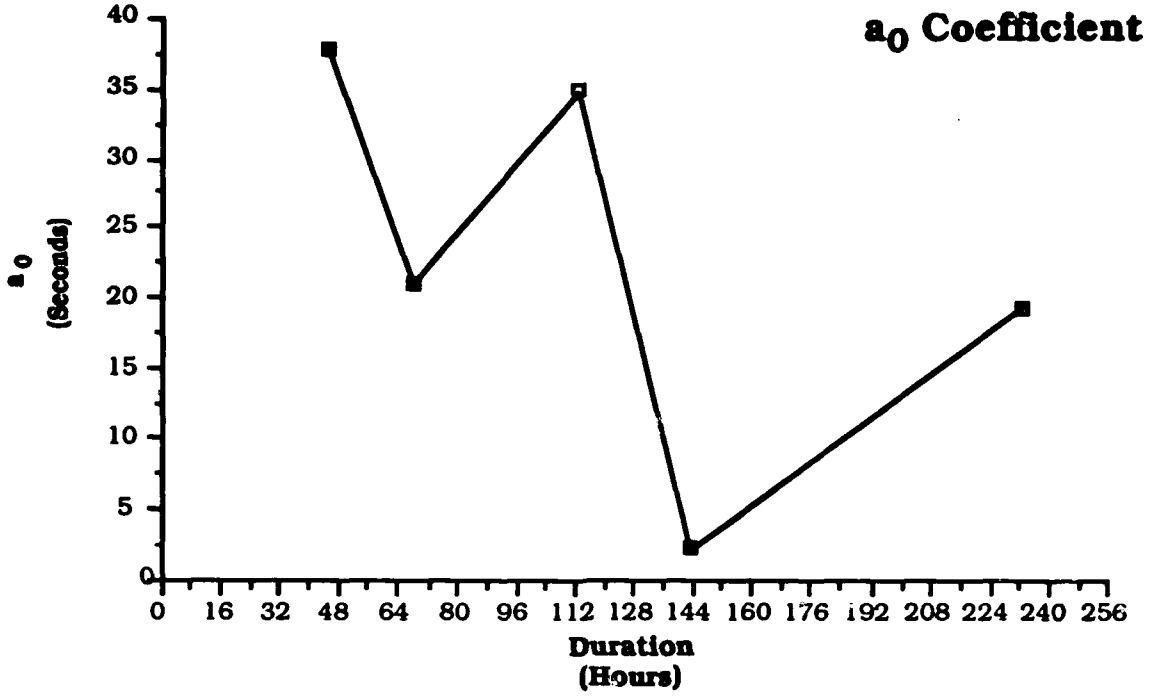
After the data was gathered from the experiment, the clock rate is calculated. The first step is to enter the data into the polynomial fitting program and the coefficients for the polynomial are calculated (see Figure 5B). This is done for all five experiments in each group. The coefficients are then analyzed to determine the actual coefficients for the global time transformation.

Coefficients

To determine the actual coefficients, the coefficients are grouped by term (i.e. a_0, a1, a2, a_3 and a_4) and plotted. Figures 5C through 5G are the plots for the groups of coefficients for Computer 2.[4] Figure 5C is a plot of the a_0 coefficient for Computer 2. The plotted values are the calcu-

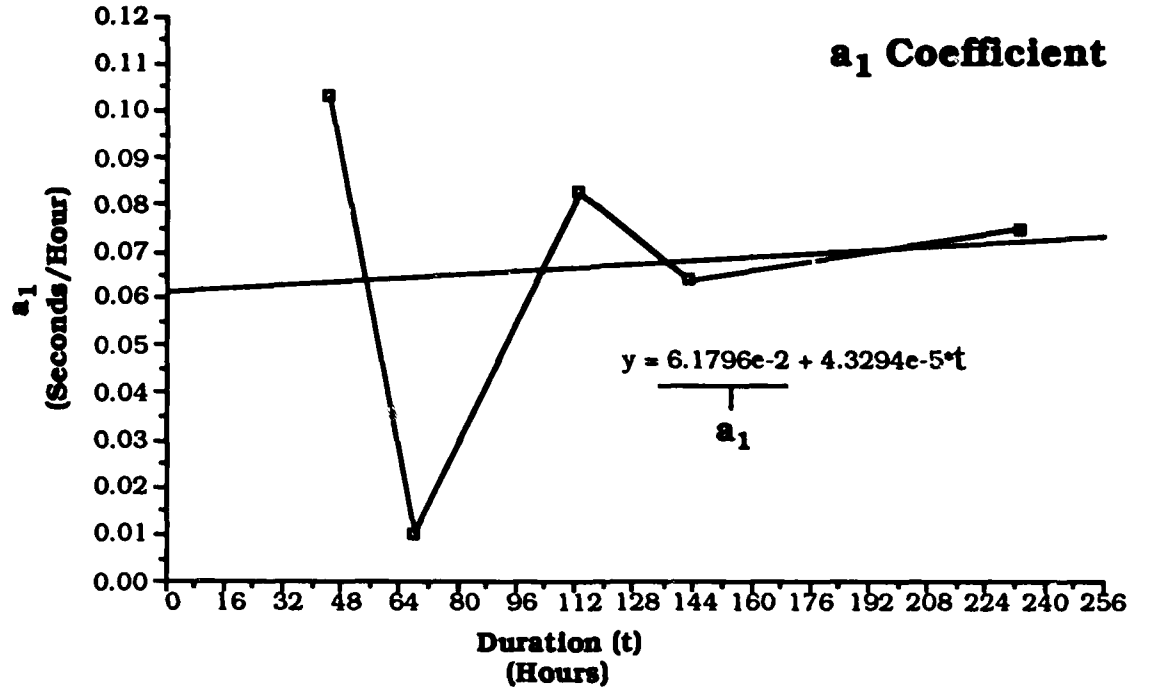
[4] Since the three systems have similar results, only the Computer 2 results will be discussed.

lated offsets at the beginning of the experiment. The measured values for the offset dependent on when the experiment is started which is random.



a_0 Coefficients for Computer 2

Figure 5C



a_1 Coefficients for Computer 2

Figure 5D

Figure 5D is the plot of the a_1 coefficients and is the most important plot of all. The main thing to notice is that the difference between successive Plot points gets smaller as the duration gets larger. This implies convergence as the duration period gets larger. This is primarily due to the fact that the variations in the message propagation become relatively small when compared to the duration time (see Equations 15 and 20 in Section 3).

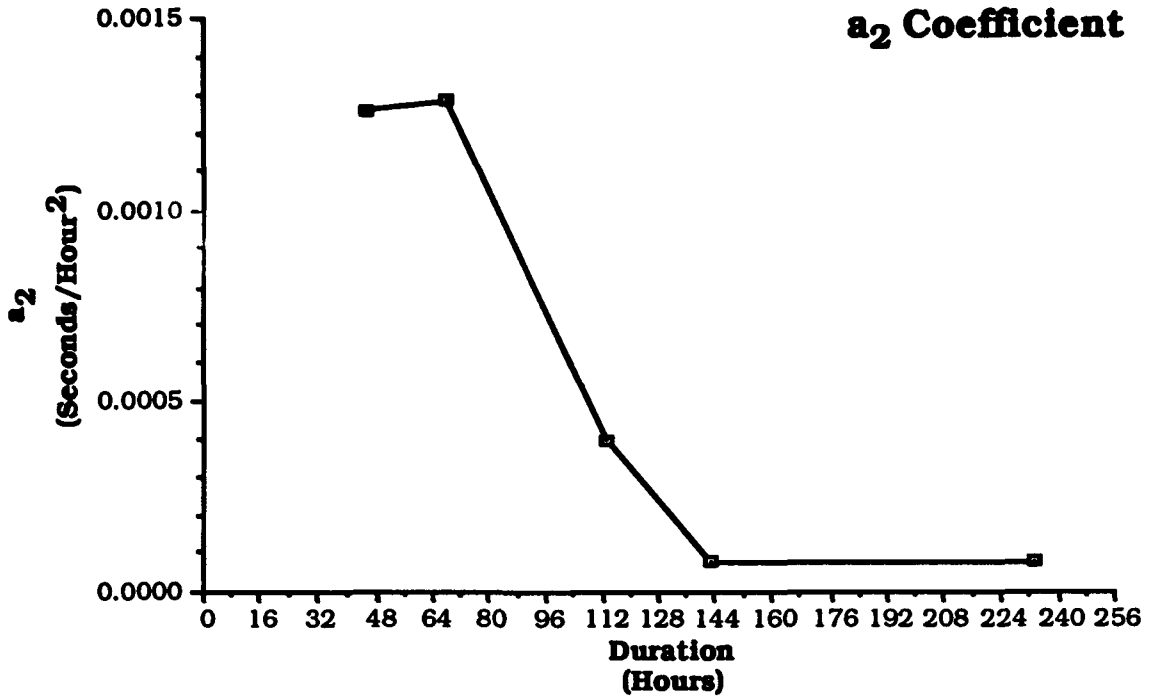
Higher Order Coefficients

Figures 5E, 5F and 5G are the plots of the a_2, a_3 and a_4 coefficients respectively. Each of these plots are similar in shape. They all start at high value and then drop off to a low value as the duration gets larger. From each of the graphs, the value for the coefficient is very small after 144 hours of duration. The sharp drop off for a_1, a_2 and a_3 would imply that *experiments have to be at least 144 hours long to provide a reasonably good estimate for the coefficient a_1 (i.e. the clock rate).*

The differences in successive plot points for the short duration values is large compared to the differences for the long duration values. The large differences in the shorter duration values when grouped in with the values for the longer duration values introduce an error into the calculation of the a_1 coefficient. The question is how to compensate for the large differences in the short duration values. Since the probability of having the correct a_1 is not the same for both the short duration values and the long duration values, a weighting function can be used. The weighting function would make the contribution of the short duration values to the calculation of a_1 smaller then the longer duration values. A probability weighting function could be applied to the analysis to give a more precise estimate. This approach was not examined in this effort. The effects of the large variations in the short duration values on the calculation of a_1 will be visible in the comparison later in this section.

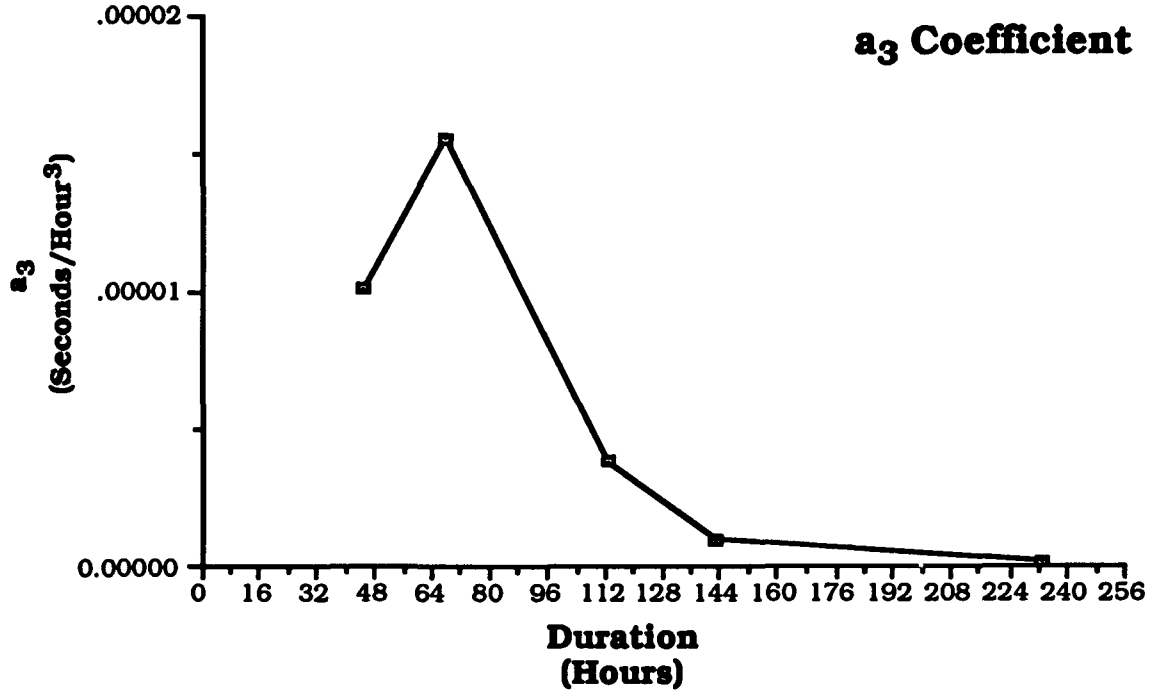
After plotting the coefficients, the rest of the numerical analysis is performed to calculate the effective value for the coefficients. The analysis is done by applying the least squares fitting program to each set of coefficient data. Like the polynomial fitting program, the least squares fitting

program generates the coefficients of a line that "fits" the data. The coefficient for the zero order term in the equation for the line is the clock



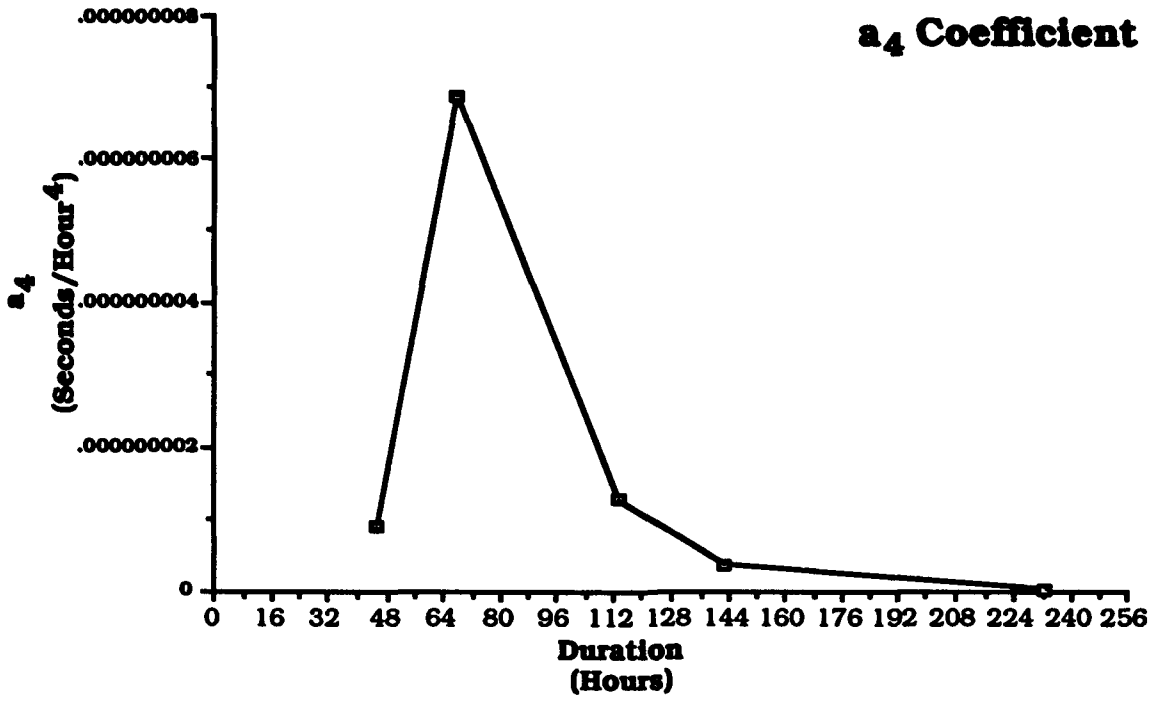
a_2 Coefficients for Computer 2

Figure 5E



a_3 Coefficients for Computer 2

Figure 5F



a_4 Coefficients for Computer 2

Figure 5G

rate (see Figure 5D). The first and higher order terms for the a_0 plot provide insight as to how precise the coefficient value is and the relative size of the error is in the calculation. If the higher order coefficients are relatively small, then the a_1 coefficient is good and the error is small. The higher order terms also implies that the duration of the experiment should be greater than 144 hours.

To determine the particular time transformation for this experiment, a clock offset is needed . The clock offset for the global time transformation will come from the first offset measurement in the second set of measurements. They both must have the same initial clock offset if a comparison between the global time transformation predictions and the measurement is to be meaningful.

Predictions versus Measurements

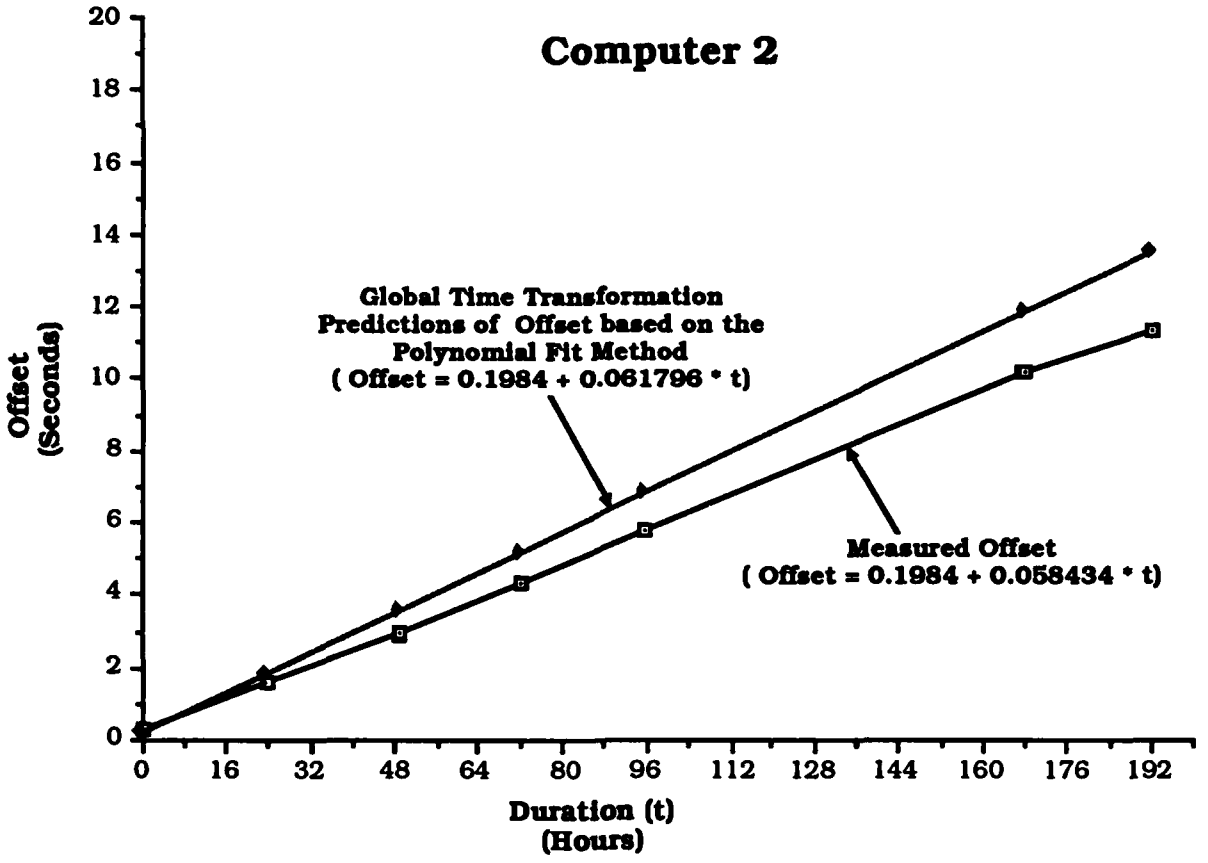
With the first set of measurements completed and the global time transformation defined, the next step is to obtain the second set of measurements. The second set of measurements is a collection of clock offset measurements that have been taken on each system over a long interval

of time. Selecting the proper measurement technique is a critical issue in taking the measurements. For the experiment, the static message technique was used because the LANPAs had only a static message capability. Even though Computer 2 has the ability to do dynamic messages, the measurements for Computer 2 were taken with the static message technique. The primary reason is that we wanted to make sure that the technique did not have any impact on the results. Therefore, the static message technique was used for both the Computer 2 and both LANPAs' measurements.

After the second set of measurements was taken, the initial clock offset measurement was used to define the particular global time transformation. The *particular* global time transformation for this experiment is

$$G(t) = 0.1984 + 0.061796 * t$$

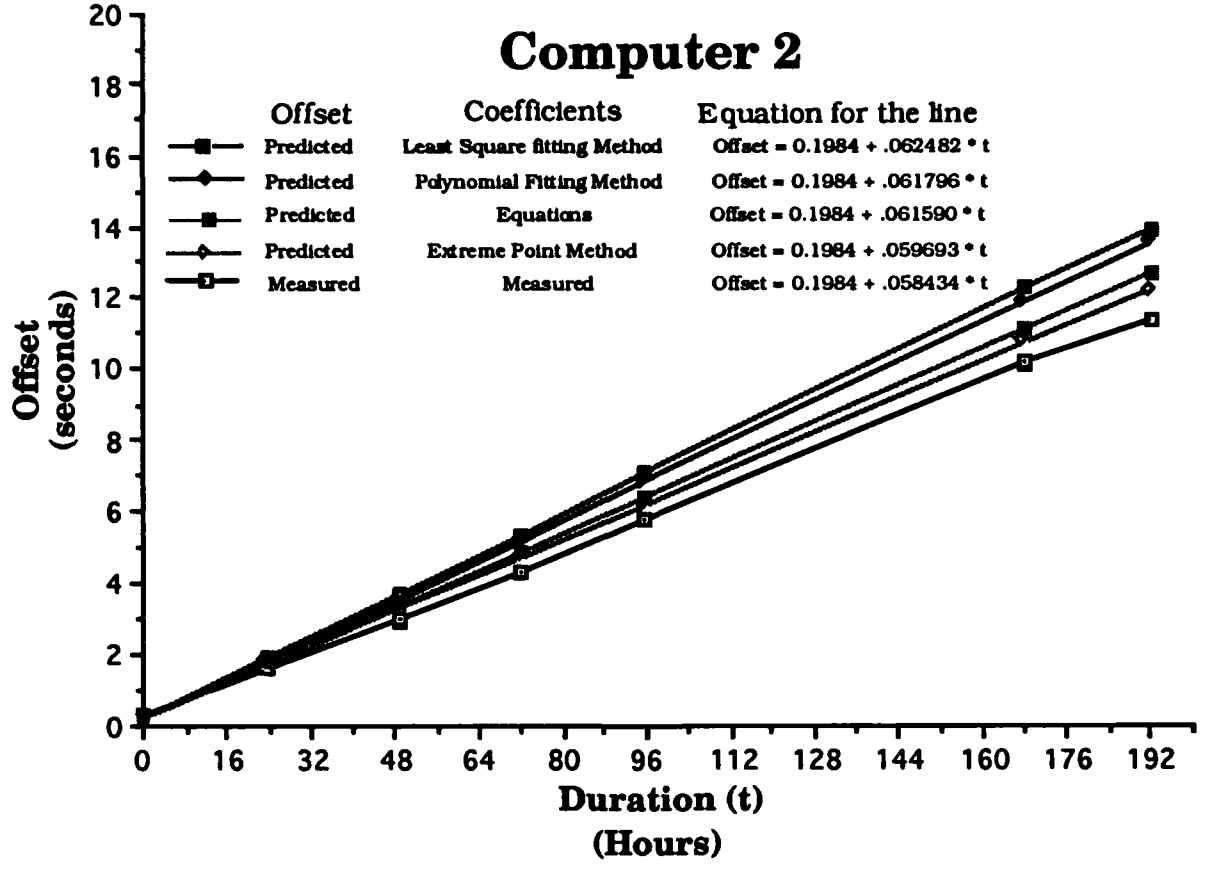
The clock offset measurements for Computer 2 and the predicted offsets from the global time transformation are shown in Figure 5H.



Predicted Offsets and Measured Offsets for Computer 2
Figure 5H

Comparison of Other Global Time Transformations

The two lines track each other relatively well. To get a better feel for how well they track each other, let's look at the results from the other three methods (see Figure 5I). The least squares fit method is the same as the polynomial fit method except this method uses the least squares fit for the first step of the calculation procedure. The resulting line is the furthest from the measurements. Another method is the extreme point method. This method takes the initial measurements and the final measurements and directly calculates the coefficient. This method produced the best approximation to the measurements. This is the same method that would be used if the analyst had a long time for measurements. The last method uses the equation derived in Section 3 (i.e. Equations 7, 15, and 20) to calculate the coefficients. Which of these equations to use depends on which end of the network the device is on and the device type.



Comparison of the various methods

Figure 5I

If the device is on the same side of the network as the global system, then use Equation 7. If the device is on the other side of the network and is a monitoring device, then use Equation 15. If the device is a computer system on the other side of the network, then use Equation 20.

Figure 5I has some interesting points. The order of the methods in terms of closeness to the measured clock offset is in agreement with one's expectations. The extreme point method for durations of this size should be the best because the impact of the variation in the message propagation would have the least effect on the value. Since the extreme point method has no higher order coefficient values, the errors due to them are eliminated. The next best result should be the equations method. The equation method is tailored to the fundamental problem because it is derived from the problem description (i.e. the interconnectivity model). The curve fitting method would naturally be last because it deals only with the data analysis and is not tailored to the problem. The large variations in the data for the short duration points produces errors in calculating the clock rate via the curve fitting method.

Refinements to each of the above methods, except the extreme point method, will generate a closer bunching of the lines. To refine the curve fitting method, the analyst can restrict the values to only the longer duration data points or introduce probability based weighting functions. The equations method can be enhanced by developing new approaches for determining how to isolate and incorporate into the calculation additional variables that effect the message propagation (e.g. device response times, network utilization and adaptive routing). New methods for measuring the clock offset may provide a better set of coefficients for the global time transformation.

The global time transformations that were generated and compared with the measured offsets were close. As stated earlier, an exact global time transformation is not possible. The goal of the global time transformation is to transform the data with enough accuracy to satisfy the analysis requirements.

Section 6: Recommendations

The global time transformation work that was discussed in this report came from the information gathered during the execution of an in-house effort. Since the primary intention of the in-house effort was to do research and not to develop global time transformations, there is a lot of work that still has to be done. This section will discuss some of the work.

The global time transformation is a deterministic function that maps the measured time data into the global time data. The particular global time transformation is obtained by calculating the coefficients for the transformation. To calculate the coefficients for the transformation, multiple experiments are run from which multiple sets of coefficient are generated. From the multiple sets of coefficients, the global time transformation coefficients are calculated using analytical methods[5]. Since there is a set of candidate coefficients to be analyzed, there is an uncertainty associated with the coefficient calculations. To effectively handle this uncertainty, the coefficient should be calculated using the theories and analytical techniques of probability. Since the measurement environment is probabilistic in nature, the probabilistic approach should provide better values for the coefficients than the deterministic approach. using probability to determine the global time transformation is an area that requires additional research.

A different technique that may improve the precision of the coefficients calculations is to have multiple global clocks. With multiple global clocks, there would be multiple sets of coefficients for calculating the coefficients. Multiple coefficients calculated by multiple observers would enable the analyst to compare a wider set of observations. The coefficients can be numerically analyzed not only within the set of values of a single observer but across a set of observers. The global time transformation predictions can be analyzed to check for consistency amongst the set of global time transformations, to see if the order of events is preserved and to minimize the prediction error of the global time

[5] In section 5, four of these methods for calculating the coefficient were discussed.

transformations. This technique should enable the analyst to determine more precisely the proper set of global time transformations.

The experiments that were implemented during this effort covered only a few months. To completely understand the characteristics of the clock rate, more extensive measurements and experimentation has to be performed. The measurements and the experimentation should examine the change in the clock rate under more rigorously controlled conditions. The two types of effects that should be examined are the physical and the operational effects. The physical effects are those changes in clock rate due to changes in the physical environment (e.g. temperature). Operational effects are those changes in the calculated clock rate due to changes in the operational environment (e.g. system loading and system interconnectivity). The basic objective of this research is to determine what and how much does the physical and operation environment impact the calculation of the coefficients and the predictions of the global time transformation.

The next two recommendations focus on keeping the clocks more aligned with each other. The first technique is to use global time transformations to predict clock offsets and then make the system adjust the clock rate to the predictions. The idea behind this approach is to fix the clock rate via the global time transformation and make the system fit the expected clock rateor global clock rate. For the global time transformation not to corrupt the data, the adjustment process has to be uniform in time.

The other recommendation is to use an expert system approach which monitors NTP time messages and then makes adjustments to the clock. The expert system continuously observes a sequence of time messages and learns from the messages when and how much the clock should be adjusted. The expert system continues to monitor the messages until the expert system can predict the occurrence of the messages. For example, if someone tells you every hour that your watch is off by 15 seconds. After a while you can *predict* when they will tell you that your watch is off and by how much it is off (i.e. 15 seconds). Once you know when to adjust your watch and by how much, the adjustment can be made regularly without the other person telling you every hour. Depending on the magnitude in the offset, the monitoring can be adjusted

to be less frequent (e.g. from every hour to every day or month). If the adjustment is spread uniformly over time, the clock will be more like a *perfect clock* than those systems that don't have this capability.

To supplement this global time transformation work, an error analysis must be done. In this effort, the error analysis was not rigorous. An error analysis becomes more important when the data resolution requirements become more demanding. The demand for more data resolution occurs when the number of events that occur in a given interval gets very large. In today's concurrent multiprocessor environments, millions of events occur in a matter of seconds. The global time transformations must be able to transform the time data well enough to distinguish the events. The precision of the transformation will depend on how much the numeric spectrum is occupied by the error bandwidth in the global time transformation. To determine the resolution of the global time transformation predictions, the analyst must determine the magnitude of the error associated with the calculation of the coefficient and with the global time transformation predictions. For the global time transformation technique to become more effective, more work has to be done on error analysis.

Section 7: Conclusion

The computer environments of today are no longer contained in the same mainframe or room or even building. Today's computer environments are multiprocessor high performance computers that interact with each other via high speed communication networks. These environments bring with them not only improved performance and functionality but also a new breed of complex analytical problems. There are many facets to the problems. One of these facets is that a time domain analysis in one of these environments is a computational nightmare. The problem is that the data is generated from multiple systems with different clocks. If the data was generated from the same clock the problem would be greatly reduced.

There are two approaches to solving this problem, the hardware approach and the software approach. The hardware approaches focus on the alignment and synchronization of the clocks. These approaches are either too expensive or are not physically feasible because the hardware uses chip technology and manufacture information is proprietary. The software approaches are better because they don't have to deal with the physical restrictions of the environment but deal only with the data, which makes these approaches less costly and easier to implement.

The global time transformation technique is one of those software approaches. The basic premises of the global time transformation approach are that there are no *perfect clocks* and that the time data comes from clocks that are out of alignment and out of synchronization. The global time transformation is a very simple technique that provides very good results. Fortunately, the transformation is a linear function which makes data transformation significantly easy to implement.

Since it is not possible to correct the underlying causes of the multiple clock system problem, the global time transformation corrects the effects of the problem. Approaches that deal with the effects generated by multiple clock systems will have a higher success rate then those that try to correct the cause of the problem. The global time transformation deals only with the effects of the problem. The work

discussed in this report only deals with some of the techniques for generating a global time transformation. There is room for new techniques to be developed and room for improvement to the techniques mentioned in this report.

This report has shown that global time transformations are linear and quite precise. To generate the global time transformation, an analyst has techniques available to him/her that can be used to measure data and to calculate the transformation coefficients. When the coefficients are substituted into the global time transformation, the analyst will be free to analyze multiple clock systems in the same way as he/she analyzed single clock systems.

The global time transformation is a candidate solution to the analyst's multiple clock data problem.

References

[Crow90] Crowcroft, J. and Onions, J. Network Time Protocol(NTP) over the OSI Remote Operations Service. RFC-1165, June 1990.

[Duda87] Duda, A., Harrus, G., Haddad, Y. and Bernard, G. Estimating global time in distributed systems, Proceeding of the 7th International Conference on Distributed Computing Systems, September 21-25, 1987, pp. 299-306.

[Kope87a] Kopetz, H and Ochsenreiter, W. Clock synchronization in distributed real-time systems. IEEE Transaction on Computers Vol. C-36, No. 8, August 1987, pp. 933-940.

[Kope87b] Kopetz, H and Ochsenreiter, W. Interval measurements in distributed real time systems. Proceeding of the 7th International Conference on Distributed Computing Systems, September 21-25, 1987, pp. 292-298.

[Lamp78] Lamport, L. Time clocks and the ordering of events in a distributed system. Comm ACM, Vol. 21, No. 7, July 1978, pp. 558-565.

[Lamp85] Lamport, L and Melliar-Smith P.M. Synchronizing clocks in the presence of faults. Journal of the ACM, Vol. 32, No. 1, January 1985, pp. 52-78.

[Marz83] Marzullo, K. and Owicki, S. Maintaining the time in a distributed system. Proceedings of the 2nd ACM SIGACT-SIGOPS Symposium on the Principles of Distributed computing , August 1983, pp. 295-305.

[Milh45] Milham, W. Time & Timekeepers. New York:The MacMillian Company, 1945. pp 197-212.

[Mill89a] Mills, D. Internet time synchronization: the Network Time Protocol. RFC 1129, Network Working Group, October 1989.

[Mill89b] Mills, D. Network Time Protocol (Version 2) Specification and Implementation. RFC-1119, Network Working Group, September 1989.

[Mill92] Mills, D. Network Time Protocol (Version 3) Specification and Implementation and Analysis. RFC-1305, Network Working Group, March 1992.

[Ofek87] Ofek, Y and Faiman, M. Distributed global event synchronization. International Conference on Distributed Computing Systems, September 21-25, 1987, pp. 307-314.

[Smar91a] Smari, W. Clock synchronization techniques: a survey. Final Report AFOSR Summer Facility Program, August 9, 1991.

[Smar91b] Smari, W. An overview of clock synchronization techniques. Technical Report AFOSR Summer Facility Program, August 1991.

[Smar87] Srikanth, T. and Toueg, S. Optimal clock synchronization. Journal of the ACM, Vol. 34, No. 3, July 1987, pp. 626-645.

*U.S. GOVERNMENT PRINTING OFFICE: 1994-510-117-50014

MISSION

OF

ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.