Presented at the 13th International Conference on Parallel and Distributed Computing: PDSC2000, Aug. 8-10, 2000, Las Vegas, NV

Time Synchronized Measurements in Cluster Computing Systems

Alan Mink, Robert J. Carpenter[1] and Michel Courson[2],
Information Technology Laboratory
National Institute of Standards and Technology (NIST)[3]
Gaithersburg, MD 20899, USA
{amink@nist.gov}

Abstract: We describe hardware time synchronization instrumentation that we have developed which achieves time synchronization of better than one microsecond. The purpose of this instrumentation is for Quality of Service characterization of parallel and distributed computing and of network communications. Using this instrumentation we measure the accuracy achieved by the NIST Autolock time synchronization algorithm, built upon the well-known Network Time Protocol (NTP) time exchange, and the various factors contributing to its performance for both local and remote computing clusters. We show that such software algorithms, once their parameters are tuned for the expected delays, can achieve accuracy close to single digit microseconds.

Keywords: Autolock, Cluster Computing, GPS, MultiKron, NTP, Time Synchronization

1 Introduction

Systems composed of multiple computers often require that all the participants have a common, synchronized view of time. The time synchronization accuracy needed depends on the application. For our PC based computing cluster environment, our application is performance measurement of parallel programs. Performance measurement can require a wide range of time synchronization accuracy, depending on the event being measured. For example, measuring the time between the processes in two different nodes reaching a synchronization point may require 1 µs, or even sub µs accuracy. Measuring communication latency between nodes connected via a high-speed switch may require accuracy on the order of 10 µs. Measurement of process execution times may require a range of accuracy from 1 µs to 100 ms. In an integrated system, such as a multicomputer, a common clock is usually available. When one isn't, sometimes a common clock can be obtained from the internal network as demonstrated by Abali, et.al. [1] to achieve microsecond time synchronized measurements.

We describe hardware time synchronization instrumentation that we have developed which achieves time synchronization of better than one microsecond. The purpose of this instrumentation is for Quality of Service characterization of parallel and distributed computing and of network communications. Using this instrumentation we also measure the time synchronization achieved by the NIST Autolock [3] algorithm, built upon the well-known Network Time Protocol (NTP) [Internet RFC-1305] time exchange, and identify the various factors contributing to its performance.

2 Overview of Time Synchronization Techniques

A number of techniques exist to synchronize the clocks in distributed systems, multiple computers, or measurement support hardware. These differ in complexity, cost, precision, and the initial startup time required to achieve a given level of precision.

The NTP [4], and its alternatives [2,3], can satisfy clock synchronization requirements to within a millisecond, given a number of hours or days to synchronize. The original-equipment time base oscillator in many computer systems is not stable enough to realize the full potential precision of NTP, so that a much more stable oscillator may need to be employed. The NTP synchronization scheme requires no special hardware other than Internet access.

Radio-based systems use either the NIST WWVB 60-kilohertz low frequency (LF) signals or the Global Positioning System (GPS) satellite 1500-megahertz (UHF) signals. Synchronization to within a few tens of milliseconds after only a few minutes averaging can be obtained by special radio receivers tuned to the 60 kHz WWVB signal. A few days of averaging may improve the precision to a few milliseconds. The WWVB signals are quite weak by the time they reach the US coastlines and do not penetrate metal, concrete or masonry buildings very well. This requires the receiving antenna must be placed outdoors in many instances. WWVB does not offer coverage outside North America, but similar services are available at some other countries. The WWVB radio approach does not rely on the Internet and thus can't be spoofed by an Internet hacker. The ultimate accuracy of LF radio systems is limited by the inability to accurately predict the delay of the radio signals resulting from radio propagation variations.

[1] Contractor to NIST
[2] Visiting Scientist from U. of Maryland, UMIACS.
[3] Certain commercial items may be identified but that does not imply recommendation or endorsement by NIST, nor does it imply that those items are necessarily the best available for the purpose.

Microsecond-level synchronization (or better) can be obtained anywhere on Earth through use of the signals from GPS satellites. GPS's UHF signals do not penetrate buildings nor heavy foliage. The GPS receiving antenna must have a reasonably unobstructed view of the sky. Professional GPS-based timing systems for computers are available at costs starting above one thousand dollars. These systems only claim one-microsecond accuracy, which is achieved after a few minutes of operation. More expensive systems can achieve 100 nanoseconds or better accuracy after a much longer synchronization period. The ultimate accuracy of GPS-based systems is on the order of 20 nanoseconds, limited by the clock accuracy of the satellites themselves.

With either of these systems, a serial interface from the external radio-based hardware to the computer system can introduce many milliseconds of unpredictable delay. There are two approaches to this problem, both require modification or development of the external hardware. (1) The cheaper approach places a capture register in the external hardware, and the current time is captured in this register upon a trigger from the associated computer system. The computer system can then read the contents of the capture register through any convenient serial or parallel interface at a later convenient time. (2) A more flexible approach is to make the radio-synchronized clock register available for low-overhead reading directly by the computer system, usually through the computer's input/output bus.

3 MultiKron® Measurement Support Hardware

It is important that measurements perturb the operation of the computer or network being measured very little. NIST has developed performance measurement support hardware called MultiKron[4] [5] that reduces the perturbation to simple "writes" to memory mapped registers.

The critical features of the MultiKron, illustrated in Figure 1, are a 56-bit timestamp counter and sixteen 32-bit Performance Counter registers. Useful measurements require synchronization of the timestamp counter in all the MultiKrons in a distributed system. This means that all the timestamps must increment at the same rate, and they must either be initially reset to zero and then started at the same time, or their timestamp values noted at the same instant. The MultiKron uses the "reset and start simultaneously" approach. The MultiKron's timestamp counter can accept frequencies up to 50 MHz. No known commercial time and frequency source provides both the RESET pulse and this range of synchronized clock signals, which is why the MultiKron/GPS hardware was developed.

[4] MultiKron is NIST's registered trademark for its computer performance measurement support hardware.

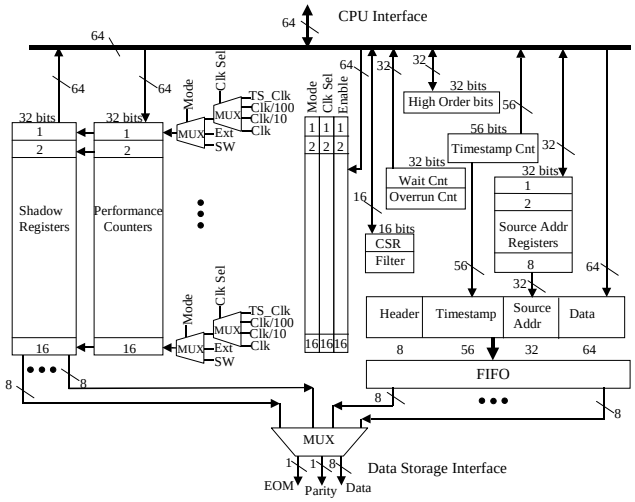

Figure 1. MultiKron_II Chip Functional Block Diagram.

Two access methods are provided to store the MultiKron's measurement data. The MultiKron Data Storage Interface port with eight data and two control bits allows direct storage of the measurement data in a dedicated memory without any processor intervention. Alternatively, the process being measured can perform low overhead memory-mapped reads of the MultiKron timestamp and the Shadow Register associated with the desired Performance Counters.

The Performance Counters are individually programmable to tally internal clock ticks or external signals via dedicated pins on the chip. They can be used to measure a number of parameters. A Performance Counter can function as a stopwatch when tallying internal clocks by enabling and disabling the Counters at the beginning and end of events. Either the CPU or the external signal can control the enable and disable. Such measurements as average frequency, average pulse duration and duty cycle can be obtained by using pairs of Counters, one for the numerator value and the other to tally elapsed time for the denominator.

3.1 Measurement Without a MultiKron

One may not need MultiKron hardware if all one needs is reasonably precise time-trace measurement without the detailed information provided by the MultiKron Performance Counters. This approach only provides a time-trace with none of the functionality of the Performance Counters. Most microprocessor chip sets contain a high-resolution cycle counter that is incremented at a multiple of the rate of an external oscillator, which is often 14.31818 MHz. This counter can be read with low overhead. A somewhat similar approach of using an external oscillator was reported by Mills [4]. Exactly the same frequency needs to be supplied to every processor in the distributed system. In order to initially correlate the time at the various points in

the system, the contents of each cycle counter must be read upon receipt of a synchronization pulse, which must occur simultaneously everywhere in the system. These synchronization requirements are very similar to those of the MultiKron.

4 Meeting Synchronization Requirements

As stated above, synchronization of MultiKron chips, or the internal counters of some microprocessors, requires synchronized frequency sources for incrementing timestamp (and other) counters and a means to simultaneously generate a synchronizing (RESET) pulse at every location, which may be widely separated. We determined that no known commercial time source provided the signals required to time synchronize the geographically-distributed MultiKrons. We also decided that the complexity of adapting these commercial GPS-based sources to our needs was similar to building our own stand-alone device. Thus we developed a system that is based on time from GPS satellites. Known commercial moderate-priced GPS-synchronized frequency sources create only 1.0 or 10 MHz, not the 50, 40, 30, 20, 15, 12.5, and 10 MHz signals required for the NIST MultiKron timestamp. Thus an external frequency synthesizer would be required even though a commercial frequency source was used.

The MultiKron RESET pulse is unique. It must be a few tens of microseconds in duration, and *terminate* at exactly the "synchronization" instant at all of the MultiKron units. This is trivial in a localized wired system, but no commercial device exists that provides such a signal for a wide-area system. GPS time sources generally provide a one pulse per second (1PPS) signal with a precise leading edge. Using the 1PPS output from a commercial unit and generating a single MultiKron-compatible RESET pulse which terminates at the correct one PPS pulse would require most of the hardware and firmware that we have used in our custom system described below.

A block diagram of the NIST GPS-based time synchronization system, MultiKron/GPS, is shown in figure 2. Operationally one MultiKron/GPS unit is used at each local site (e.g., machines in a single room). Its outputs, a stable frequency and a precision reset pulse, are distributed to each MultiKron installed in a computing node. The MultiKron/GPS contains a stable crystal oscillator to generate the signal required by the MultiKron time stamp counter. The oscillator frequency is divided down to produce one pulse per second (PPS) which is phase locked to the one PPS pulse available from a GPS receiver. The phase lock loop (PLL) corrects the frequency of the local crystal oscillator to keep the local and GPS one PPS signals synchronized to within a fraction of a microsecond. This results in excellent long-term frequency stability since the total number of locally generated cycles over a long period must equal those of the GPS satellite's clock, otherwise the one PPS signals would drift apart. However, there are short-term variations of tens or hundreds of nanoseconds in the GPS 1PPS signals due to receiver noise, Selective Availability modification of the GPS signal, and multipath propagation effects. The long-term average of the GPS 1PPS pulses is extremely accurate. As is common in GPS-based frequency or time sources, the local high-quality quartz crystal oscillator is relied upon for frequency and time stability over short time periods. The frequency of the signal produced by this oscillator is slowly corrected by the PLL. The slow correction of the local oscillator averages out the short-term variations in the GPS 1PPS signal to the extent possible given the stability of the MultiKron/GPS's crystal oscillator. GPS time, and thus MultiKron/GPS output, is synchronized with UT1 (Universal Time 1).

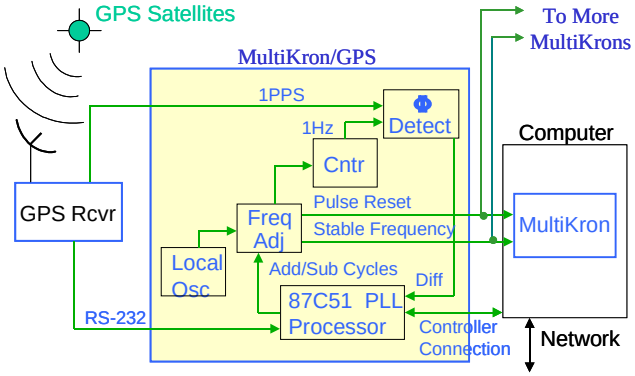

Figure 2. Block Diagram of MultiKron/GPS Time Synchronization Instrument.

An 87C51 microcomputer in MultiKron/GPS is part of the oscillator correction second-order phase lock loop (PLL) system. It also monitors the serial time data stream from the GPS receiver and commands from the attached computer. Based on the GPS signals, and the commands received for the attached measurement computer, it causes a synchronizing RESET pulse. This pulse occurs simultaneously at all the widely distributed MultiKron/GPS units and resets the MultiKron time stamp counter (clock) to zero at all sites. One should recognize that systems using a much higher quality and more expensive oscillator on the MultiKron/GPS could achieve better time stability and accuracy.

The synchronizing RESET pulse terminates at each site exactly when a local one PPS pulse occurs. The selection of which one PPS pulse to use is governed by the following protocol. Each MultiKron/GPS has a special connection to one of the machines at the local site. We designate that machine the controller. One controller notifies all the other MultiKron/GPS controllers in the distributed system via a communication network such as the Internet, in advance, of the even-numbered minute to

signal its MultiKron/GPS that a RESET is requested. Each controller monitors the low bandwidth time data stream from its GPS, and activates its MultiKron/GPS anytime within that even-numbered minute. Activation during an odd-numbered minute is ignored. The choice of a minute long activation window and a minimum of two minutes between consecutive resets are somewhat arbitrary, but was selected both for convenience and to be long enough to avoid any race conditions. Once activated, all the MultiKron/GPS units cause a RESET exactly at the 30th second of the next odd-numbered minute.

5 Network-Based Synchronization Algorithm Accuracy

The previous sections of this paper describe our measurement system and its time synchronization. It provides us with the means to evaluate software time synchronization algorithms. The accuracy of algorithms for time synchronization over communications networks is limited by the variation in the communication delays between two nodes, not the magnitude of the delays. These algorithms identify network delay and oscillator inaccuracies as limiting factors. We have identified computer system overhead as a third factor. We have developed a revised network time protocol, illustrated in figure 3, to eliminate the effect of computer system overhead. The standard time exchange protocol gets a local time stamp, t_1, at the application level and places it in a message. The message is sent down through the operating system protocol stack to the network, after traversing the network it is then received on the destination computer and again passed up through the protocol stack to the application where another local time stamp, t_2, is acquired. The computer system overhead occurs while the message traverses the protocol stack. Our revised protocol, see figure 3, modifies the operating system code and overwrites the time stamp in the message with a more current time stamp, t'_1, just before the message is transferred to the network interface card and the interrupts are disabled. Similarly on the receive side, when a message arrives and the interrupts are disabled, the receive time stamp, t'_2, is acquired and passed to the application along with the message. Thus the time interval $t'_2 - t'_1$ is an accurate measurement of the network delay without the additional overhead of the protocol stack included in t_2-t_1.

Using the MultiKron/GPS hardware described earlier, we conducted a number of experiments to measure the communication delay variation and the accuracy of software time synchronization algorithms over typical networks that would be expect to be encountered with computing clusters. The variation in communication delay provides an indication of the expected accuracy that time synchronization algorithms can achieve. We also measured the contribution of the limiting factors. We used the NIST Autolock time synchronization algorithm. The Autolock and NTP algorithms use the same NTP protocol to exchange clock information and determine the communication delays. These algorithms differ in the statistical analysis methods they use to determine the variation and how they correct for the oscillator drift. However both methods yield comparable reported accuracies.

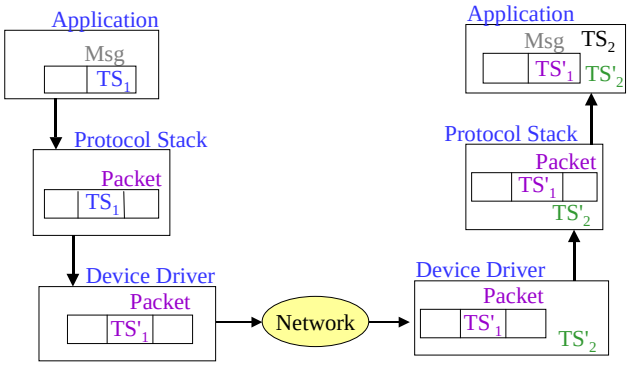

Figure 3. Revised Time Exchange Protocol.

We used our experimental PC based computing cluster interconnected via a 100 Mbits/s Fast Ethernet switch as a typical cluster environment [6]. For a remote cluster environment we used two separate clusters, about a half a kilometer apart, interconnected via a corporate intranet consisting of multiple LANs interconnected via a combination of switches and routers. These LANs consist of both 10 Mbits/s and 100 Mbits/s segments. By remote cluster we mean the dynamic integration of individual clusters over an intranet or Internet to act as a single temporary cluster. For both local and remote clusters, we instrumented two nodes with our MultiKron/GPS instrumentation. One node acted as the time reference, while the other node acted as a client trying to synchronize its local clock to the reference node time. The reference node distributed the MultiKron/GPS synchronized time. This results in a stable target time. A standard local oscillator, not the MultiKron/GPS instrumentation, drives the local clock of the software time synchronization algorithm client. The MultiKron/GPS instrumentation in the client node is used only for measurement purposes.

5.1 Cluster Measurements of the Autolock Protocol

The Autolock time synchronization algorithm requires about 12 to 24 hours to stabilize. The measurements presented below represent a small sample of traffic taken after the algorithm has reached steady state. The accuracy of these measurements is better than 1 µs. The total communication delay for each NTP protocol exchange is shown by the bar chart of figure 4(a). Each bar is divided to show the contributions of the network delay (bottom bar), the computer system send overhead (middle bar) and the receive overhead (top bar) for a quiet local computing cluster. Autolock periodically issues 5

consecutive NTP protocol exchanges. The standard deviation of the delay for each group of 5 is computed as the variance plotted in figure 4(b), which indicates the expected time synchronization accuracy. The upper set of points represents the communications variation, about 9 μs, based on the standard NTP exchange protocol. The lower set of points represents the communication variation, about 2 μs, based on our revised NTP exchange protocol. Because our revised protocol eliminates the variation of the send and receive it yields a lower variation and thus better time synchronization accuracy. We notice that the delay of the first of each group of 5 Autolock protocol time exchanges is higher than the subsequent four. This is due to caching of the code and can be eliminated if the measurement for the first exchange is discarded. If this is done, the variation results of the standard and revised protocol are similar.

We repeated this same experiment on the same system, but this time with the nodes busy running applications vs. being quiet. The communication delay plotted in figure 5(a) and the corresponding communication variation plotted in figure 5(b) show that delays and the variation are much larger in busy systems than in quiet ones. The expected time synchronization accuracy from the standard protocol is about 50 μs, while the revised protocol yields about 25 μs. Since the busy communication variation is larger than the quiet system, the effect on synchronization accuracy from caching is negligible.

We repeated this experiment on the remote cluster system, under both quiet and busy conditions. The communication delay is plotted in figure 6(a) and 7(a), for quiet and busy system respectively. The corresponding communication variation is plotted in figure 6(b) and 7(b), for quiet and busy system respectively. Due to the additional complexities of the intranet, the delays and variations of the remote cluster are higher than for the local cluster and thus also are the expected time synchronization accuracy. The expected time synchronization accuracy from the standard protocol is about 300 μs and 30000 μs, for quiet and busy system respectively. The expected time synchronization accuracy from the revised protocol is about 300 μs and 15000 μs, for quiet and busy system respectively.

The plot of figure 8 shows the error in the measured time synchronization of the quiet local cluster using the standard protocol during a 60-hour steady state period. These error measurements are taken every 20 s and are the difference between the MultiKron/GPS clock and the Autolock synchronized clock. Figure 8 indicates a time synchronization error of about 70 μs compared to the expected value of about 9 μs. The plot of figure 9 shows the error in the measured time synchronization of the quiet remote cluster using the standard protocol during a 35-hour steady state period. This represents a time synchronization error of about 1500 μs compared to the expected values of about 300 μs. The difference in these measured values and expected values is due to the default settings of the algorithm parameters that are tuned for 1 ms time accuracy.

Autolock is configured to use three parameters to describe the user's requirements for speed, accuracy and network traffic. "Sigma" defines the requested level of time accuracy. "Tconf" is the time constant for the phase-lock-loop; a smaller value speeds up the locking process but makes the system more sensitive to noise and therefore worsens its short-term stability. A time range, “Tmin” & “Tmax”, defines the interval between requests to the server. Our initial focus targeted sigma for accuracy. As shown in Figure 10, we consistently achieved accuracy under 10μs with network traffic of five 68-byte messages every 15 minutes, except for isolated glitches that we are currently investigating. These glitches are not due to interrupts and occur in 1 measurement but not in the measurement 20 s earlier and 20 s later.

6 Conclusions

We determined that standard software time synchronization algorithms, which synchronize to an external timeserver via the Internet, provided insufficient accuracy for performance measurement of local and remote computing clusters. To provide the microsecond accuracy required, we developed hardware instrumentation, called MultiKron/GPS. We based this design on GPS technology, since GPS was the only widely deployed technology that could deliver microsecond accuracy at the dispersed locations. The output of MultiKron/GPS provides a GPS trained frequency in the range of 10 MHz to 50 MHz along with a precision pulse reset signal to synchronize our MultiKron performance measurement instrumentation installed in each cluster node. These signals can be relied upon to provide considerably better that one microsecond time precision over any time interval, at least when all units are receiving the same GPS satellites.

We conducted experiments to determine the accuracy of the NIST Autolock time synchronization algorithm. We enhanced the accuracy of the Autolock algorithm by developing a revised time exchange protocol that eliminates the variance contribution of the operating system and the protocol stack. For both local and remote clusters busy running applications, the revised protocol was effective in increasing expected time synchronization accuracy. For quiet (quiescent, not running applications) remote clusters the revised protocol had little effect on expected accuracy, since the variation of the network delay far exceeds that of the operating system variation. For quiet local clusters, the revised protocol was effective in increasing expected accuracy, mainly compensating for caching effects.

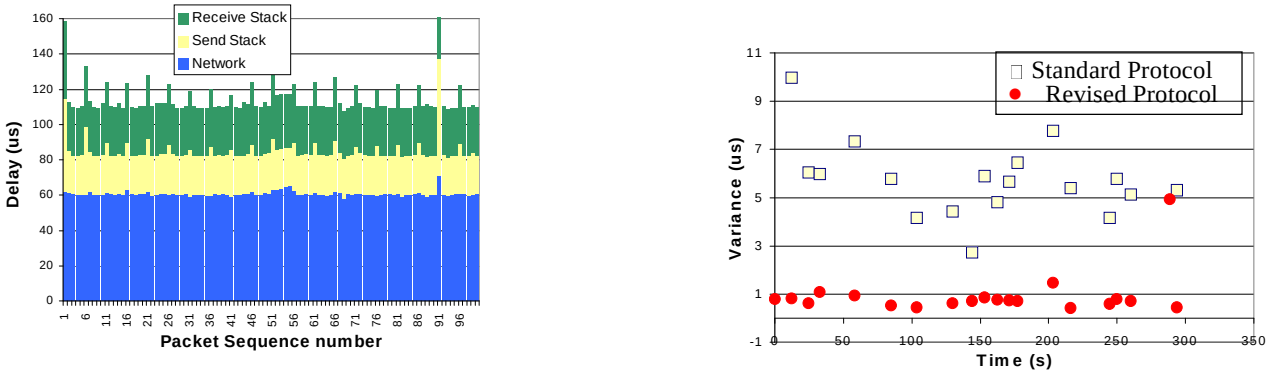

Figure 4. (a) Local Cluster Communication Delay and (b) Variation for a Quiet System.

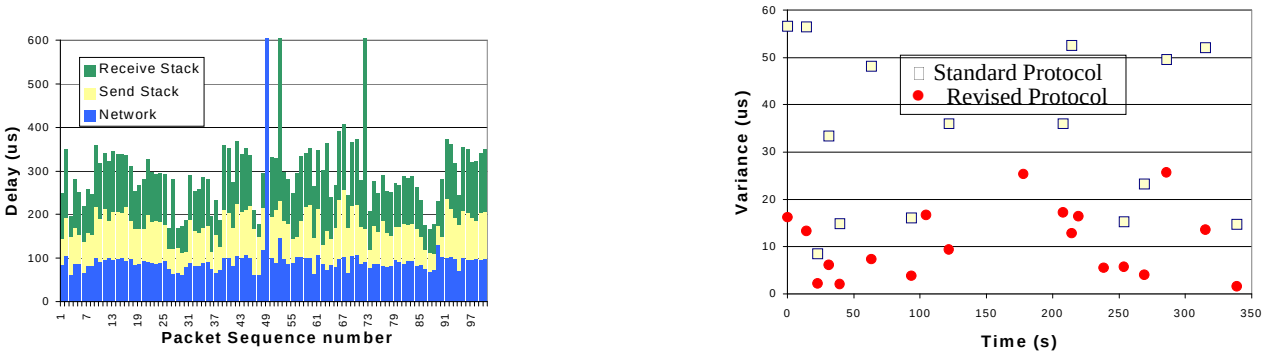

Figure 5. (a) Local Cluster Communication Delay and (b) Variation for a Busy System.

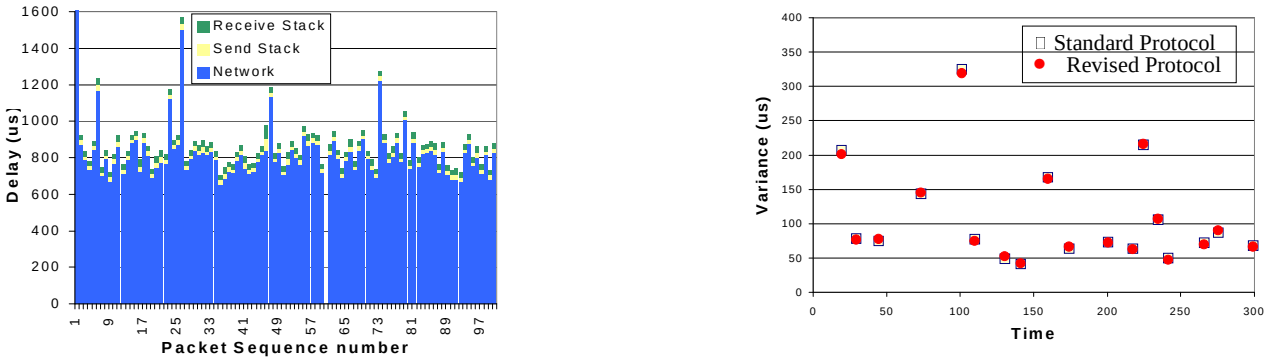

Figure 6. (a) Remote Cluster Communication Delay and (b) Variation for a Quiet System.

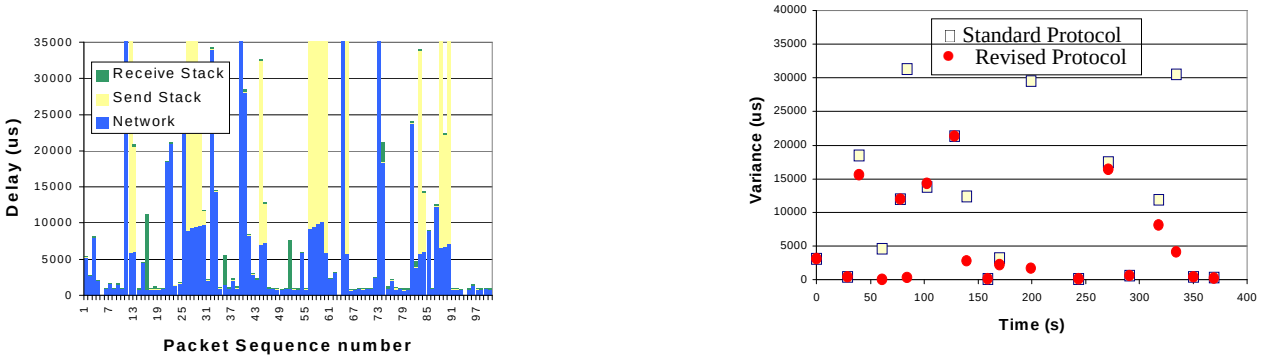

Figure 7. (a) Remote Cluster Communication Delay and (b) Variation for a Busy System.

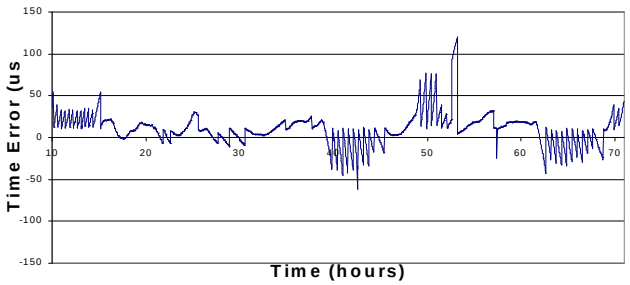

Figure 8. Measured Time Synchronization Error for the Quiet Local Cluster Using the Standard Protocol.

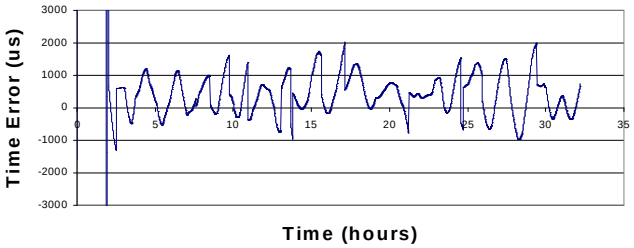

Figure 9. Measured Time Synchronization Error for the Quiet Remote Cluster Using the Standard Protocol.

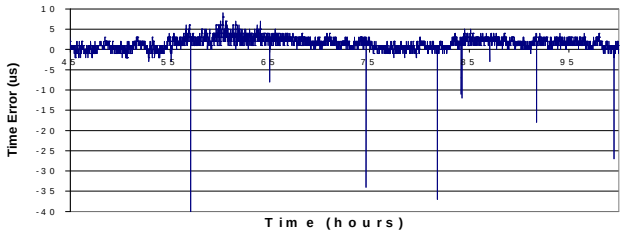

Figure 10. Measured Time Synchronization Error for the Quiet Local Cluster Using the Standard Protocol with Autolock parameters tuned for 10 microsecond accuracy.

We also determined that the measured time synchronization did not achieve its expected value. This was mainly due to the default internal parameters of the algorithm that were tuned for the more common millisecond range, rather than our targeted microsecond range. Once these parameters were adjusted, the anticipated accuracy was achieved. Our measurements further suggest that these algorithms can achieve 10 us, or better, for local cluster systems that have low variance in their network delays. This implies that hardware support may not be necessary for each node within a cluster, but only one node to act as the reference time server using software algorithms, with the revised time exchange protocol, to synchronize the other nodes.

References

[1] B. Abali, C. Stunkel and C. Benveniste, “Clock Synchronization on a Multicomputer”, IBM Research Report, Mar. 1996.

[2] J. Levine, “Time Synchronization Using the Internet”, IEEE Trans. on Ultrasonics, Ferroelectronics and Frequency Control, Vol. 45, No. 2, pp 450-460, Mar. 1998.

[3] J. Levine, “Time Synchronization Using the Internet Using an Adaptive Frequency-Locked Loop ”, IEEE Trans. on Ultrasonics, Ferroelectronics and Frequency Control, Vol. 46, No. 4, pp 450-460, July 1999.

[4] D. Mills, “Improved Algorithms for Synchronizing Computer Network Clocks”, IEEE/ACM Trans. on Networks, pp 245-254, June 1995.

[5] A. Mink, “Operating Principles of Multikron II Performance Instrumentation for MIMD Computers”, NISTIR 5571, National Institute of Standards and Technology, Dec. 1994.

[6] W. Salamon and A. Mink, “Linux Clusters at NIST”, Linux Journal, Issue 62, pp 105-109, June 1999.