29th Annual Precise Time and Time Interval (PTTI) Meeting

THE APPLICATION OF NTP TO NAVY PLATFORMS

Karen F. O'Donoghue and David T. Marlow
Naval Surface Warfare Center Dahlgren Division
17320 Dahlgren Road, Dahlgren, Virginia 22448 USA
{kodonog|dmarlow}@nswc.navy.mil

Abstract

Navy platforms consist of many systems ranging from navigation and ship control to tactical analysis, sensors, and weapons. These systems, each composed possibly of multiple computing elements, are inherently time-dependent. In addition, modern surface ships are quickly adding additional COTS based computing elements and communication networks to address emerging requirements. Vital to making such a distributed system function in this environment is a stable robust time service. The Navy has been investigating various methods for the provision of a time service on shipboard platforms. This paper presents an overview of previous Navy efforts, a discussion of the metrics identified as applicable to the Navy environment, and a summary of the HiPer-D time synchronization subnet performance. In addition, this paper identifies ongoing and future work required to provide a stable robust time service in a shipboard environment.

INTRODUCTION

The modern-day surface ship is composed of a number of separately developed and maintained systems which together execute the functions needed to operate the entire platform. These systems work independently and collectively to perform the ship's mission. A key to making such a collection of systems work in this real-time environment is a common understanding of time. Time information must be represented accurately and consistently throughout all the systems of the platform and within the various components of the individual systems. This time information includes both time of day and time interval measurements. The end result is that Navy platforms require a stable robust time service.

The Navy has been investigating the requirements placed on a time service by Navy platforms and specific time service metrics applicable to these platforms. It has also been investigating various methods for the provision of a time service on shipboard platforms. The Network Time Protocol (NTP) has been identified as a technology with the potential to meet Navy shipboard time service needs. Various studies have been conducted to analyze the performance of NTP on particular computing elements,including modification of NTP parameters to better address specific Navy shipboard requirements.

The High Performance Distributed Computing Program (HiPer-D) is a joint effort between the Navy (the AEGIS Program Office) and the Defense Advanced Research Projects Agency (DARPA) to look at the use of distributed computing technologies. The Navy efforts in the area of time services were applied to the HiPer-D testbed. The HiPer-D testbed includes a heterogeneous computing base operating in a distributed fashion over multiple network technologies to provide actual AEGIS Combat System functionality. To support this testbed, a time synchronization subnet was designed and implemented using both NTP and GPS components. The long-term performance of the HiPer-D time synchronization subnet was measured and the results analyzed. Additional efforts including specific studies and experiments for future work were identified based on the metrics defined below and the experimental observations.

GENERIC TIME SERVICE MODEL

The provision of quality time information in a computer requires that a number of functions be carried out in a coordinated fashion. This collection of functions, related to computer clocks and the time information in a computer, is referred to as a *time service.* A generic time service model incorporates all of the components necessary to provide and manage quality time information in a system. A system is a collection of interconnected computers participating in related tasks. The foundation of this time service is the clock present in each computer. Three basic components that interact with these clocks are clock coordination, clock access, and time management. A *clock coordination service* synchronizes individual computer clocks to each other and to national and international time standards (e.g. UTC). A clock coordination service includes the mechanisms necessary to exchange time information between individual computer clocks and the algorithms required for processing this information in order to arrive at meaningful conclusions. The *clock access component* of a time service addresses the provision of time information to users. This time information may include both the current time value and the accuracy of that value. These users can be human operators, application processes, or the computer's operating system. The *time management service* component of a generic time service includes the functionality necessary to monitor and control both the various computers' clocks and the clock coordination service within a synchronization domain or system. One example of a clock management service is monitoring the accuracy achieved by the clock coordination service in a particular computer and modifying the operational parameters if the accuracy does not meet specified bounds.

TIME SERVICES FOR NAVY PLATFORMS

Navy ships are designed to perform a number of complex realtime tasks ranging from navigation and ship control to controlling complex sensors and weapon systems. This collection of applications places rather stringent requirements on the supporting computing and communication infrastructure, including the time service. First, to provide a stable and robust time service, clocks need to be synchronized in both phase and frequency to a defined standard (e.g. UTC). The Navy (via the Next Generation Computer Resources Program) has identified a basic phase accuracy requirement of 1 ms. [3]. Further analysis is necessary to look at the requirements of new and emerging applications.

There are a number of additional requirements for time services on Navy platforms, especially in light of the rapid evolution of shipboard platforms, tactical systems, and the technologies on which these are based. First, the time service needs to be fault-tolerant, surviving failures of both the time servers and the interconnecting communication links. Secondly, new platforms under development are demonstrating a trend to distribute processing to an increasing number of computers aboard a surface ship. As more processors are added, the time service must be scaled to meet new requirements. Experience has shown that techniques which adequately synchronize two computers are not necessarily sufficient for dozens or hundreds of computers. The time service must allow the incorporation of additional components requiring synchronization, with minimal impact to the existing time service and ship infrastructure. Finally, the time service needs to be cost-effective, utilizing the existing communication infrastructure and Commercial-Off-The-Shelf (COTS) components where possible. Concepts discussed here may have counterparts in commercial systems, particularly specialized applications such as air traffic control, but in general the requirements go beyond those found in non-military applications. Specific requirements of shipboard time services are discussed in detail in [3].

EXISTING TIME SERVICE APPROACHES

There are a number of existing approaches for clock synchronization on surface ships. The classic approach is the use of a single master clock chassis that distributes time information to a limited set of computers via a set of point-to-point computer interface channels. More recent approaches have included the use of specialized clock interfaces located within computers and synchronized via a separate physical infrastructure using timecodes (e.g. IRIG-B). There is often a backup master clock or time source with a duplicate distribution network to support it. There are a number of common themes to these approaches to clock synchronization. These approaches strive to distribute time information to the computers rather than synchronize the computer's own clock resources. No attempt is made to coordinate the existing clock resource with the synchronization source. Secondly, these approaches utilize a special distribution network for time information. This involves a separate physical plant for the distribution of time information. Scalability issues arise with the time service as the number of computers increases.

APPLICABLE METRICS

Based on the needs of Navy platforms, time service metrics specifically applicable to these platforms have been identified. The first metric of interest is the most commonly recognized, *phase accuracy*. Clock phase represents the time value of the clock, and the accuracy associated with it is the difference between the phase of the clock of interest and the national or international reference standard. This is analyzed by measuring the clock offset between a clock and a reference source at any given instant in time. The second metric of interest is *frequency stability*. This represents the difference between the frequency of the clock of interest and the national or international standard. This is analyzed by observing the clock offsets between a clock and a reference source over a given interval of time. This is most commonly represented using the Allan variance. [4] The third metric of interest to the Navy is the *fault recovery time*. Navy platforms must adapt quickly to realtime conditions which may include failed communication links and time servers. In addition, the impact of these failures must be minimal. The time required to detect and recover from a fault condition is,therefore,defined as the fault recovery time. A final metric is *clock settling time*. This metric represents the time it takes the time service to reach a defined phase accuracy in an individual computing element after the machine begins initialization. These metrics help to characterize the time synchronization needs of Navy platforms. Engineering choices are made regarding resources consumed versus performance obtained. The trade-offs made by Navy platforms will be different than those of the global Internet.

THE NETWORK TIME PROTOCOL

The Network Time Protocol (NTP) [1] is a distributed clock synchronization protocol that provides for the coordination of interconnected computer clocks utilizing the existing communication infrastructure. NTP was developed by Dr. David Mills at the University of Delaware for use in the Internet community. NTP estimates the phase and frequency offset between two peer clocks and provides corrections for use by the local clock routines. NTP is a connectionless time information exchange protocol and an associated set of algorithms to process and act on the time information collected by the protocol. Some of the key features of NTP are highlighted here. First, NTP is based on a returnable time approach that reduces impact of communication path delays. The NTP synchronization subnet is a self-organizing hierarchy of time servers. Redundancy can be incorporated in the synchronization subnet using multiple servers and clock selection algorithms. NTP provides both phase and frequency corrections for the local clock. Finally, NTP operates in connectionless mode using the User Datagram Protocol (UDP) and the Internet Protocol (IP) in order to minimize latencies, simplify implementations, and provide ubiquitous internetworking.

NTP PARAMETER ANALYSIS

Initial experiments using NTP on small realtime networks produced peak clock phase offsets on the order of a few milliseconds using COTS equipment and default parameters for the public domain distribution of NTP available from the University of Delaware. The default parameters for NTP are tuned for operation in the global Internet. As part of the Navy's analysis of NTP, it was determined that one promising path was the tuning of various NTP parameters to provide a responsiveness acceptable for a military system. Simulations and analysis indicated that the NTP client polling interval could be set to a shorter interval to provide faster synchronization, greater stability, and more rapid detection of any fault conditions. By default the NTP polling interval varies between a minimum of 64 seconds and a maximum of 1024 seconds depending on the perceived stability of the time synchronization subnet. The 1024-second default maximum polling interval is generally sufficient for the vast majority of applications. Only networked systems running applications that have strict clock synchronization requirements would have a need for a maximum polling interval smaller than 1024 seconds (i.e. a shipboard combat system). In such systems, shortening the maximum NTP polling interval may provide improvements in the quality of clock synchronization and the responsiveness of the synchronization subnet. There is a basic engineering tradeoff to be made between the amount of improvement in synchronization accuracy and the increase in network traffic and CPU processing required.

A number of experiments have been performed at the Shipboard Network Technology at the Naval Surface Warfare Center Dahlgren Division (NSWC-DD). Some of these experiments focused on the impact of lowering the maximum polling interval for NTP. The purpose of these experiments was to determine the effect of changing the maximum polling interval (from the default of 1024 seconds to 64 seconds) on the quality of clock synchronization between the NTP clients and the local time server. Several computer platforms were used in this experiment. The platforms were synchronized to a local time server using NTP with either the default (1024) or the modified (64) maximum polling intervals. The offsets were measured once a minute over a period of several hours. The results were examined to determine the amount of improvement in clock synchronization.

Examples of the results are shown in Figure 1. These plots show the offsets measured for both maximum polling intervals (1024 and 64 seconds) on a number of Sun workstations. Two separate data sets are overlaid to show the performance of the same platform utilizing either of the two polling intervals over several hours. The polling interval change has improved clock synchronization by about a factor of ten. Using the default polling interval settings, the clock offsets would wander as high as a few milliseconds. By shrinking the polling intervals to 64 seconds, the clock offsets are kept in the range of hundreds of microseconds. Additional experiments were done on polling intervals of 32 and 16 seconds with little appreciable improvement in clock synchronization accuracy. This analysis is specific to individual hardware and operating system combinations and may need to be revisited as systems evolve.

THE HIPER-D SYNCHRONIZATION SUBNET

The Department of Defense has directed the use of COTS components using commercial standards wherever possible. In addition to this general directive, there has been ongoing research into innovative distributed computing architectures for traditionally federated systems such as the AEGIS combat system. In this context, the High Performance Distributed Computing Program (HiPer-D) was conceived as a joint experiment, teaming the AEGIS Program Office with DARPA. The purpose of the HiPer-D experiment was to explore the feasibility of inserting DARPA-developed distributed computing technologies into the AEGIS combat system.

The HiPer-D experiment performs a significant number of the key functions of the AEGIS combat system, focusing on the Anti-Aircraft Warfare (AAW) problem space. The HiPer-D system is built using a fully distributed fault-tolerant architecture. The computing plant used for this system is a heterogeneous mix of systems from different vendors. In addition to the various hardware platforms and operating systems, four different network technologies are used, including Ethernet, FDDI, ATM, and Myrinet. Figure 2 illustrates the hardware and network configuration. Table 1 summarizes the platform and operating systems used.

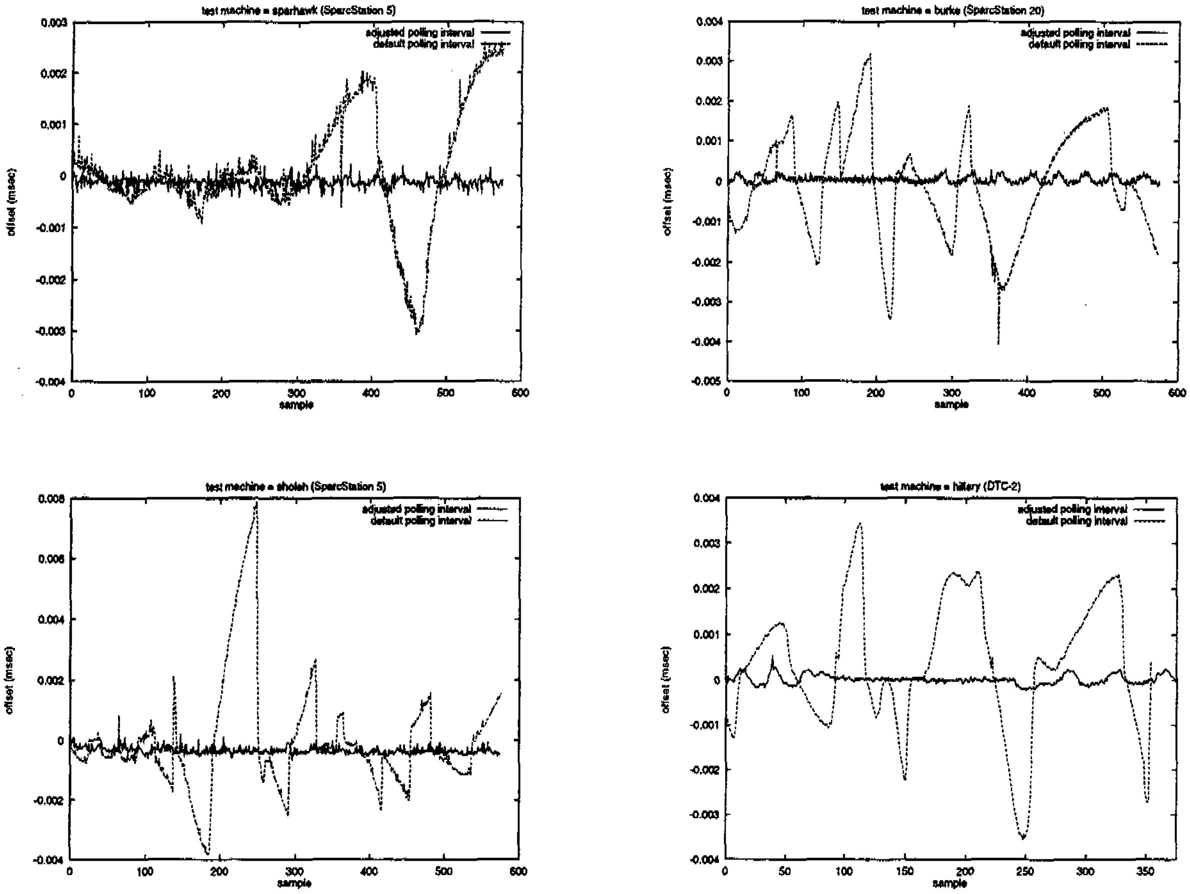

Figure 1: Sample Data from NTP Maximum Polling Interval Analysis

Table 1. HiPer-D Platform Configurations

Vendor	Hardware Platforms	Operating Systems
DEC	Alpha 200 4/233 Alpha 3000/600 Sable SMP 4	OSF/1.3.2
HP	HP 9000/770 (TAC-4)	HP-UX 9.07 HP-UX 10.10 OSF-RT PA
SGI	Origin 2	IRIX 6.3
SUN	SPARCstation 10 SPARCstation20 Ultra1 Ultra2 Sun 4/630	Sun Solaris 2.5.1
	Pentium PCs	OSF-RT

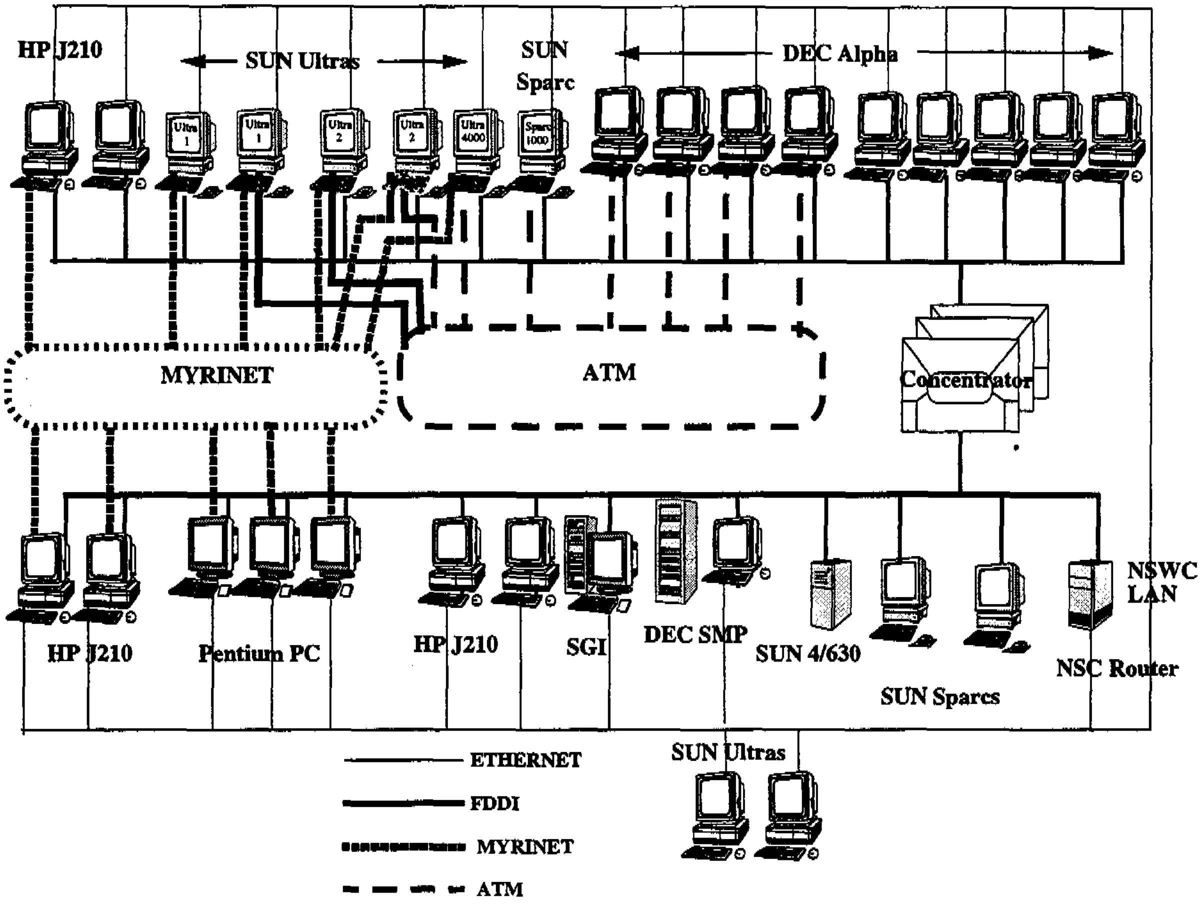

Figure 2: The HiPer-D Testbed Configuration

Based on initial experiments involving NTP in the Shipboard Network Technology Lab, the focus of the HiPer-D time service effort was incorporating improved time synchronization capability into the HiPer-D configuration. In the HiPer-D testbed many technology and operational prototypes are integrated together to study larger system-wide effects and capabilities. Here, the time service is one necessary component within the prototype shipboard system.

A time synchronization subnet was designed to meet the goals of the HiPer-D testbed. First, the synchronization subnet was to provide a stable time service for the HiPer-D experiment using the available COTS components. Second, the synchronization subnet was to require minimal modification of the available COTS products. Finally, due to the way the HiPer-D lab operates, the synchronization subnet was to operate with minimal operator intervention. The synchronization subnet was to be viewed as a component of the computing infrastructure. The goal was to determine the level of performance obtained based on the above objectives and constraints. Additionally, it was anticipated that the HiPer-D testbed could improve on the 1 millisecond requirement identified previously by Navy studies. The synchronization subnet designed is shown in Figure 3. There are four NTP client configurations. The primary time server is using GPS (via IRIG-B) and its own local clock as potential synchronization sources, with the understanding that the local clock will only be used in case the GPS interface fails. The backup servers each peer with the primary server and their own local clocks. All clients peer with the primary server and the two secondary servers. It is expected that all machines will be using the primary time server as their synchronization source. This is not viewed as an optimal synchronization subnet, merely a sufficient one given the resources at hand. The polling interval is set to once every 64 seconds for all platforms. Most platforms had multiple network interfaces, and, in these cases, time information exchange is normally conducted using the default network interface.

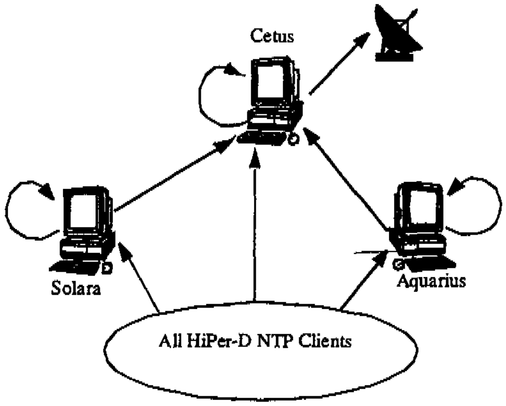

Figure 3: HiPer-D Synchronization Subnet Configuration

PERFORMANCE

The goal of this performance study was to analyze the long-term phase accuracy of the synchronization subnet, independent of any specific activity on the systems including CPU load and network load. Data werecollected every fifteen minutes for a two-month period from September 3, 1997 to November 10, 1997. No limitations were placed on testbed activity during this timeframe. Table 2 shows data for each individual platform. Figure 4 illustrates the performance obtained for a few specific platforms. This performance shows that synchronization well below a millisecond was achieved on the vast majority of the platforms. There were some anomalies that require further investigation. The older DEC Alpha platforms performed much more poorly than newer Sun and HP platforms. This is to be expected given the rapid advances workstation vendors have made in the area of local clock hardware and software over the last few years. Note, this experiment only analyzed the first metric, phase accuracy. Additional data and analysisare required to examine the three remaining metrics.

SUMMARY AND CONCLUSIONS

Analysis of the data discussed above provides some initial conclusions. First, the synchronization subnet described performed quite well. Newer generations of machines demonstrated greater phase accuracy than older generations. Not surprisingly, classes of hardware running the same operating system demonstrated similar behavior. The reduction of the maximum polling interval proved to be one mechanism to improve the phase accuracy. Further investigations could reveal additional mechanisms. HiPer-D made an engineering trade-off that was acceptable for that particular environment. This trade-off may not be appropriate in more resource constrained systems.

Table 2. HiPer-D Synchronization Subnet Platform Specific Results [1]

Machine Name	Avg. Offset	Std. Dev.	% samples less than .1 (ms)	% samples between .1 and .5 (ms)	% samples between .5 and 1 (ms)	% samples between 1 and5 (ms)	% samples greater than 5 (ms)
DEC[2]							
andromeda	1.154	0.847	5.459	23.883	19.805	50.853	0.000
aries	1.010	0.775	7.319	25.182	22.965	44.534	0.000
auriga	1.213	0.851	4.901	20.068	19.712	55.304	0.016
callisto	0.936	0.577	3.784	18.781	37.919	39.516	0.000
columbia	0.659	0.531	12.498	36.254	25.306	25.942	0.000
gemini	0.960	0.761	8.559	26.066	24.128	41.247	0.000
hydra	0.911	0.661	6.575	24.376	33.726	35.323	0.000
leo	0.671	0.535	12.704	34.388	26.105	26.803	0.000
taurus	0.669	0.739	12.390	35.432	25.880	26.283	0.016
titus	0.984	0.801	4.823	27.636	27.900	39.625	0.016
HP							
crux	0.116	0.421	84.210	12.399	1.149	2.129	0.113
hercules	0.071	0.132	85.802	12.804	0.737	0.658	0.000
jots1	0.292	0.242	38.413	25.917	35.554	0.116	0.000
jots28	0.068	0.111	80.290	19.157	0.252	0.300	0.000
myra	0.079	0.214	82.346	16.608	0.499	0.499	0.048
norma	0.056	0.099	89.556	9.885	0.341	0.217	0.000
vela	0.092	0.162	70.959	27.744	0.479	0.818	0.000
SGI							
capella	0.087	0.458	71.546	28.076	0.221	0.079	0.079
SUN							
aquarius	0.102	0.063	44.070	55.853	0.031	0.046	0.000
aquilla	0.641	2.117	0.217	7.953	91.364	0.435	0.031
blofeld	0.604	0.123	0.248	7.881	91.219	0.652	0.000
carina	0.121	0.088	41.891	57.891	0.125	0.093	0.000
mercury	0.105	0.092	54.544	44.913	0.465	0.078	0.000
pavo	0.160	0.106	27.679	71.499	0.744	0.078	0.000
phoenix	0.610	0.089	0.016	4.746	94.960	0.279	0.000
saturn	0.473	0.091	0.140	62.872	36.864	0.124	0.000
solara	0.108	0.847	57.875	41.387	0.569	0.154	0.015
venus	0.090	0.106	65.545	33.432	0.807	0.217	0.000
PENTIUM PC							
ceres	0.451	0.263	8.364	50.083	41.289	0.265	0.000
luna	0.448	0.232	4.488	54.737	40.593	0.182	0.000
pallas	0.537	0.240	1.573	48.394	49.901	0.132	0.000
vesta	0.492	0.270	5.929	44.303	49.619	0.149	0.000

---

[1] Samples taken every fifteen minutes from 2200 on September 3, 1997 to 1515 on November 10, 1997.

[2] The DEC platforms are the oldest machines in the HiPer-D configuration representing an older generation of hardware and software. It is believed that newer generation hardware and software would perform on par with the other workstation vendors.

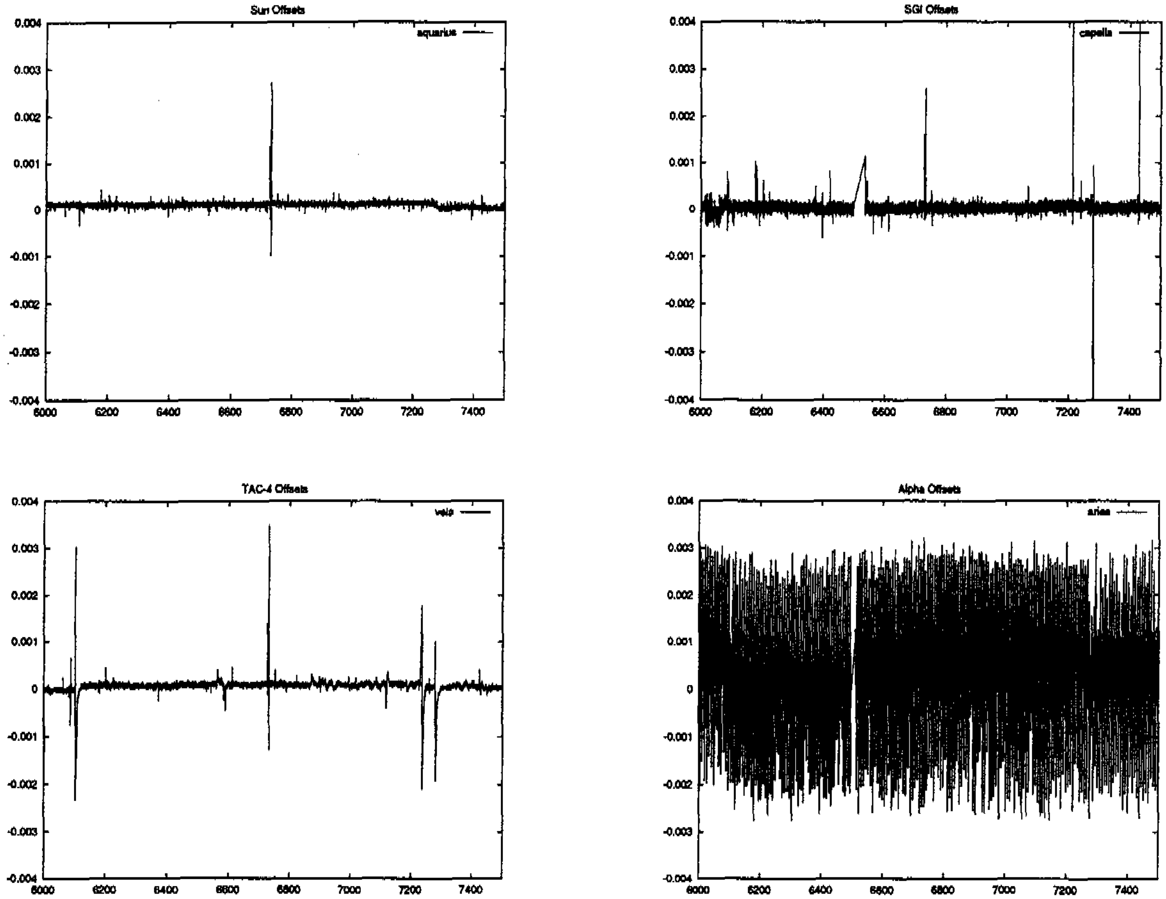

Figure 4: Sample HiPer-D Synchronization Subnet Platform Performance

A number of items have been identified as future work efforts. First, at least one and preferably two additional time servers need to be added to the HiPer-D testbed. A synchronization subnet that contains multiple time servers could provide greater fault tolerance. Performance analysis of multiple servers, especially in the case of server failure, is necessary. Second, although the quality of clocks in COTS workstations has improved over the past few years, new kernel modifications based on research by Dr. David Mills [2] are beginning to be delivered with commercial products. The Navy is depending on the COTS products to provide future generations of computing infrastructure for its platforms. Analysis of the workstation products being delivered in general and of the specific new kernel features in particular is necessary. Time service performance is tied closely to specific clock implementations provided by computer vendors. The Navy will need to test new generations of hardware and software to ensure that they provide the needed functionality. Finally, this study looked at the long-term phase accuracy of the HiPer-D synchronization subnet. Analysis of the frequency stability based on the collected data has not been completed at this time. In addition, no data were collected to study the fault recovery time or clock settling time metrics. Finally, focused analysis of the synchronization subnet under specific stress or fault scenarios (cpu load, network load, server failure, etc.) would be useful. All identified metrics need to be appropriately analyzed for the HiPer-D testbed.

The Navy has invested in the analysis of time services in general and NTP in particular. Results have shown that this technology, in conjunction with modern computing equipment, can provide a stable robust time service for Navy platforms.

ACKNOWLEDGEMENTS

A number of individuals contributed to the completion of this cycle of data collection and analysis. In particular, Barbara Davis assisted in the maintenance of the synchronization subnet, Timothy Plunkett conducted the NTP polling interval experiments and produced some of the analysis tools, and D. Wayne Mills provided review and oversight.

REFERENCES

[1] D.L. Mills 1992, "Network Time Protocol (Version 3) specification, implementation and analysis," RFC 1305.

[2] D.L. Mills 1996, "The Network Computer as Precision Timekeeper," 28th Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting, pp. 97-107.

[3] K. F. O'Donoghue and D. T. Marlow, "Time Synchronization Services Aboard Surface Ships," Naval Engineers Journal, Vol. 108. No. 6., pp. 73-84.

[4] D.W. Allan, Characterization of Clocks and Oscillators, ed. D.B. Sullivan, D.W. Allan, D.A. Howe, and F.L. Walls, NIST Note 1337, 1990, pp. TN121-TN128.