5-2007

Application of High-precision Timing Systems to Distributed Survey Systems

Brian R. Calder
University of New Hampshire, Durham, brian.calder@unh.edu

Rick T. Brennan
NOAA

C Malzone
Reson, Inc

J Marcus
Reson, Inc

Peter Canter
Applanix Corporation

Application of High-Precision Timing to Distributed Survey Systems

Brian R. Calder[1], LCDR Richard T. Brennan[2], Chris Malzone[3], Jon Marcus[3] and Peter Canter[4]

Abstract

In any hydrographic survey system that consists of more than one computer, one of the most difficult integration problems is to ensure that all components maintain a coherent sense of time. Since virtually all modern survey systems are of this type, timekeeping and synchronized timestamping of data as it is created is of significant concern. This paper describes a method for resolving this problem based on the IEEE 1588 Precise Time Protocol (PTP) implemented by hardware devices, layered with some custom software called the Software Grandmaster (SWGM) algorithm. This combination of hardware and software maintains a coherent sense of time between multiple ethernet-connected computers, on the order of 100 ns (rms) in the best case, of the timebase established by the local GPS-receiver clock.

We illustrate the performance of this techniques in a practical survey system using a Reson 7P sonar processor connected to a Reson 7125 Multibeam Echosounder (MBES), integrated with an Applanix POS/MV 320 V4 and a conventional data capture computer. Using the timing capabilities of the PTP hardware implementations, we show that the timepieces achieve mean (hardware based) synchronization and timestamping within 100-150 ns (rms), and that the data created at the Reson 7P without hardware timestamps has a latency variability of 28 µs (rms) due to software constraints within the capture system. This compares to 288 µs (rms) using Reson's standard hybrid hardware/software solution, and 13.6 ms (rms) using a conventional single-oscillator timestamping model.

1 Introduction

Data collected or constructed in multiple computers connected into a survey system can only be integrated if there is some common frame of reference between the systems. Since it must be inherent to the structure of the data, this frame of reference is typically time. Unless special care is taken, however, each computer system will develop its own sense of time locally through a combination of local oscillator and real-time clock chip. Data timestamped according to these disparate systems will require latency corrections before they can be merged, which may be difficult to determine with sufficient accuracy to avoid significant attitude- or position-induced bathymetric artifacts. As an alternative, the survey system can be configured to transfer data from the computers where it is generated to a single capture station that applies timestamps from a single oscillator. The latency between local computer clocks is avoided, but the transmission of data between systems incurs transmission latencies which may be very variable depending on transmission mechanism and system loading. In either case, it is clear that careful control of timekeeping protocols is required to maintain data quality.

A number of approaches to this problem have been proposed (Figure 1). They vary considerably in terms of their quality of time synchronization, complexity of implementation and cost.

The single-oscillator model (Figure 1(a)) has essentially no control over latency in transmission between the components of the system, and at the same time channels all data into one system. This concentration of data results in high processor loads at the capture station, which exacerbates any problems with interrupt latency for external data; latency in communications, particularly in ethernet switches, can be considerable and extremely variable with data volume. As data volumes increase with new MBES systems

[1] Center for Coastal and Ocean Mapping and NOAA/UNH Joint Hydrographic Center, University of New Hampshire, Chase Ocean Engineering Lab, 24 Colovos Road, Durham NH 03824, USA. (brc@ccom.unh.edu)
[2] Hydrographic Systems and Technology Program, NOAA's National Ocean Service, 1315 East West Highway N/CS11, Silver Spring MD 20910, USA. (richard.t.brennan@noaa.gov)
[3] Reson, Inc., 100 Lopez Road, Goleta CA 93117, USA. (malzonec@reson.com, jmarcus@reson.com)
[4] Applanix Corporation, 85 Leek Crescent, Richmond Hill, Ontario, Canada L4B 3B3. (pcanter@applanix.com)

and more systems use ethernet for transmission of data, these problems become proportionately more significant.

The distributed serial message model (Figure 1(b)) resolves the transmission latency issue by transferring a sense of time from the GPS-aided Inertial Motion Unit's (IMU's)[1] GPS-receiver clocks to the data generation systems, allowing them to timestamp the data directly; this timestamp is used in post-processing rather than that applied by the data aggregator (typically the capture computer). Timekeeping internal to the IMU is typically linked to the internal GPS receivers, and is typically on the microsecond scale for most modern systems. The mechanism used for transfer of the timebase, however, results in latencies since serial lines are not designed for low or even constant latency transmission; the latency is lower and more stable than transmission of data to a central computer, but this limits the precision of time synchronization achievable. Stability also relies on the PC clock, which is not especially stable or reliable. In addition, to transfer time to more than one receiver, the serial line output must be split but cannot be buffered, since that would cause additional latency variability. Most serial lines have limitations on their output drive capabilities, and splitting the signal results in complex cabling requirements with signal degradations. With multiple participants, the ability of the component computers to maintain time synchronization depends strongly on the properties of their serial hardware and synchronization software; both are not well controlled. This solution has been shown to achieve synchronizations on the millisecond scale [1], and is currently used on most NOAA platforms [2], but is not inherently scalable.

An alternative scheme to simple distribution by serial message is to construct a software/hardware hybrid, Figure 1(c), where base timestamps are distributed by serial line to the system's PC clock, while the high-precision timebase signals (generally a one pulse-per-second (1PPS) hardware signal from a GPS-receiver clock) are used to trigger a hardware counter that provides the least-significant digits of the timestamp. This scheme provides good synchronization with low latency (believed to be within the microsecond range, although real-world tests of this have not been published before now) due to the hardware signals, although it has the same limitations on fan-out, drive capacity and cabling difficulties as the Distributed Serial Message model of Figure 1(b), making it difficult to scale to more than a few systems. This scheme is currently used by Reson 7P processors such as the one used in these experiments.

The distributed timebase model (Figure 1(d)) resolves most problems of latency and scalability. The timebase is transferred to a hardware device in one of the component computers, typically via a coaxial cable. This signal defines the short-term seconds tick, but does not provide the epoch (i.e., absolute time) information; this is provided by the serial line carrying an appropriately formatted message once per second. The hardware device skews its local oscillator to tick at the rate and the instant prescribed by the timebase transfer, and attributes the ticks with the serial message data in order to define an absolute time timepiece with accuracy as good as the short-term stability of the local oscillator (typically a temperature- or oven-controlled quartz oscillator). This timebase is then re-distributed to similar hardware devices in all of the other components of the system using a proprietary format and specialist cabling; details vary between types of hardware device with the most common being the IRIG-B format. The timing of the serial epoch messages is not critical, and the remainder of the synchronization occurs in hardware with very low (and very stable) latency, leading to high-precision time synchronization at the level of the hardware devices. Re-distribution of the timebase requires some significant effort in cabling and configuration management, although it is more scalable than other solutions. The accuracy is known to be in the microsecond range [3], and although systems of this type are in routine use with some survey systems [4], they have not been widely adopted.

In general, the ideal solution would be one with time synchronization under 1ms (rms), and preferably on the order of a few tens of microseconds; low complexity of implementation, and especially ease of expansion to multiple participant computer systems; and low total cost of implementation, factoring in hardware costs and cost of maintenance and configuration management. Judged against these criteria, all of the currently proposed solutions have deficiencies. This paper examines the behavior of an alternative system which is a type of distributed timebase model, except that the message distribution for the timebase from the hardware devices occurs over conventional ethernet connections, and may be done over the same ethernet connection as is used for data transfer between the component computers in the survey system.

[1] The term 'IMU' correctly refers only to the motion sensing unit which measures angular rates and linear accelerations that are then processed to provide attitude information. We use the term here, however, in its common usage as generic shorthand for the assemblage of motion sensor component, GPS receivers and computer that make up the composite attitude and positioning system commonly used in survey operations.

We build on top of the IEEE 1588-2002 Precision Time Protocol standard [5], which provides basic hardware synchronization, with custom software called the Software Grandmaster (SWGM) algorithm [6], which provides epoch information to correct the PTP times to UTC times. In addition to the benefits of a distributed timebase model as described above, this method provides for essentially infinite scalability, since all that is required is a sufficiently large ethernet switch, and since the distribution mechanism is simple ethernet the cable and configuration management costs are the same as simply connecting the component computers into the survey system in the first place.

The remainder of this paper outlines the algorithms used in distributing the timebase, and the experiments used to determine the properties of the timestamps derived from this distribution in a standard modern survey system. We show that the system generates very high-precision timestamps, that the complexity of the implementation is low, and that the algorithm performs better than available alternatives.

2 The Precision Time Protocol and Software Grandmaster Algorithms

2.1 Precision Time Protocol

The IEEE 1588-2002 Precision Time Protocol (PTP) was designed for the distributed instrumentation and control markets, with the goal of extending a consistent timebase between multiple computing devices using only the ethernet cables that were used to connect them. In the protocol, one timepiece in the network of connected devices is chosen to become the master device, and distributes its sense of time (derived from a local well-controlled hardware oscillator independent of the main computer's oscillator) across the ethernet to all other subordinate participants. Inherently, however, ethernet is a non-deterministic protocol, with no guarantee of the length of time that a packet takes to cross the network from one device to another. If there are several switches between the source and receiver, it is possible for any one packet to be delayed for an arbitrary period of time, or even to be lost completely. This unknown latency results in a stochastic timebase adjustment being added to the time developed at the master timepiece as the information flows across the network to the subordinates, leading to unpredictable data timestamp latencies in the survey system.

The latency is not completely random, however. Unless there are significant changes in the network structure, or very bursty network traffic, the latency between two endpoints is typically relatively stable; if this latency can be estimated, then time corrections between the clocks can be implemented. The messages sent from each clock, Figure 2, contain the time of transmission of the message, determined with as little latency as possible from the hardware PTP oscillator. The latency from master to subordinate timepiece can therefore be computed directly by subtracting the timestamp contained in the message from the time of reception, giving an observed latency of:

$$\begin{aligned}\Lambda_{MS} &= t_S - t_M \\ &= (t_M + \epsilon + \lambda_{MS}) - t_M \\ &= \epsilon + \lambda_{MS}\end{aligned} \tag{1}$$

where λ_{MS} is the true latency between the two ethernet transceivers, and ε is the (unknown) offset between the local oscillators at either end of the connection. Similarly, the observed latency from the subordinate to the master timepiece can be determined from the Delay Request ('What time is this?') message sent from subordinate to master by subtracting the transmission and reception times:

$$\begin{aligned}\Lambda_{SM} &= t_R - t_T \\ &= (t_T - \epsilon + \lambda_{SM}) - t_T \\ &= -\epsilon + \lambda_{SM}\end{aligned} \tag{2}$$

where λ_{SM} is the true latency between the two transceivers, going from subordinate to master. The average true one-way latency is therefore readily seen to be recoverable from the mean of the observed latencies:

$$\begin{aligned}\lambda_{EQ} &= (\Lambda_{MS} + \Lambda_{SM})/2 \\ &= (\epsilon + \lambda_{MS} + [-\epsilon + \lambda_{SM}])/2 \\ &= (\lambda_{MS} + \lambda_{SM})/2\end{aligned} \tag{3}$$

and consequently the offset between subordinate and master timepiece as seen from the subordinate timepiece can be directly computed as $\epsilon \approx \Lambda_{MS} - \lambda_{EQ}$. This observed offset error can be used as an input to a control loop to discipline the local oscillator at the subordinate timepiece so that it follows the master oscillator directly, effectively driving the observed offset to zero over time. In practice, the subordinate oscillators are driven to match the master oscillator with an offset error that reflects the degree to which the latency varies with respect to the update time of the algorithm's control loop (typically 0.5Hz) and the symmetry of the transmission path latency (since the computation of the average one-way latency assumes that the path is symmetric).

The success of the algorithm depends on the precision with which the transmission and reception times of the ethernet messages can be determined. In most PTP implementations, this is done in hardware at the ethernet physical interface, so that any jitter in the upper levels of the ethernet protocol stack do not affect the measurements. The requirement for very precise measurements require the use of a 'follow up' message, since the PTP implementation does not know the true time of transmission until the message is actually sent. Implementations capable of generating the 'follow up' message fill in an approximate time in the first message, and then the precise time generated from the first message into the second. This is not mandatory in the standard, but significantly improves the degree of synchronization. It is possible to implement the PTP algorithms without hardware assist, using a standard computer oscillator and normal ethernet cards. Software-only PTP can achieve reasonable synchronization performance [7], although this is limited by the degree to which the software can be integrated into the ethernet driver's protocol stack within the operating system. In effect, this limits software-only PTP with precision acceptable for hydrographic survey to Linux style operating systems where such integration is possible. Since most survey platforms run on Windows-based devices, this is not a viable solution.

2.2 Software Grandmaster Algorithm

The PTP algorithm achieves the synchronization of all subordinate timepieces to the master timepiece. It does not, however, guarantee any synchronization or syntonization (i.e., matching in frequency and phase) of the master timepiece to a standard timebase, such as UTC. The PTP standard allows for an external device, termed a 'Grandmaster Clock,' to provide this guarantee. Within a survey system, however, it is important not just to synchronize to an arbitrary timebase, but to ensure that all devices are synchronized to the same timebase. Since the GPS receivers within the survey system develop a sense of time linked to the atomic clocks on the satellite vehicles as part of their normal processing, it makes sense to base all timekeeping against this standard rather than an external device. In fact in most modern survey systems the motion sensor, or Inertial Motion Unit (IMU), is GPS aided and the GPS timebase is used internal to the device for its timestamps on attitude and positioning data; by making the IMU the timebase generator, therefore, we can guarantee that all timestamps on bathymetric or sidescan data match the position and attitude timestamps exactly and directly without further processing for latency corrections.

The ideal configuration for this would be for the IMU to support PTP directly on its ethernet output port. If the IMU disciplined its PTP oscillator to UTC time using the GPS receiver clocks and became master timepiece in the timing network, this timebase would be automatically distributed across the ethernet, and all subordinate timepieces would agree on UTC time to within the precision of the synchronization (Figure 3(a)). In practice, no current generation IMU has this capability, and significant time will be required to implement it. In order rapidly to investigate the performance of such a system, we implement here a version of the Distributed Timebase Model, using the PTP ethernet cards to distribute the timebase, and disciplining one of the PTP timepieces to the UTC epoch information (in software) and synchronization 1PPS signals (in hardware) in order to provide a stable timebase (Figure 3(b)). We call this the Software Grandmaster algorithm, since it is functionally equivalent to a PTP Grandmaster Clock, except that it synchronizes to a particular timebase rather than developing its own, and involves no further hardware devices.

The SWGM algorithm operates in multi-threaded mode, Figure 4, having one thread to monitor the hardware at the master timepiece and distribute epoch information to the rest of the network, and one thread to monitor the epoch information at each subordinate timepiece (as well as at the master) in order to provide appropriately corrected timestamps to the user-level code. The PTP hardware is configured to record, at a maximum precision of one PTP oscillator clock cycle, the time at which the 1PPS event is delivered to the master timepiece by the IMU. The transmission thread monitors the serial ports on the host computer for the master timepiece and records the approximate PTP time and precise (actually ideal) UTC time contained in the messages from the IMU containing the epoch information. Pairing the results of these activities allows the master timepiece to develop a correspondence table between PTP and UTC times, which it then multicasts to all subordinate timepieces using the same mechanism as the base PTP algorithm, assuring that the correspondences will always be distributable if the PTP algorithm is working. The reception thread at the subordinate timepieces records a short buffer of the correspondences locally, and builds statistics from these and the locally observed offset from the master timepiece in order to calibrate corrections to its local oscillator's PTP time (which is maintained in synchronization with the master timepiece's time through the underlying PTP algorithm) and thereby provide UTC timestamps to the local user. From the user's point of view, however, the complexity of correction is hidden behind a simple API so that timestamps are simply generated on demand with respect to UTC time, including an estimate of the uncertainty of the timestamp based on the observed correspondence table, local offset and software loading factors. Full details of the SWGM algorithm can be found in [6].

2.3 Integrated Reson SWGM Application

In order to test the SWGM algorithm in an operational setting, integration with a MBES data capture system was required. We integrated the software developed at CCOM/JHC with the data capture system for the Reson 7P MBES topside system, and utilized a National Instruments PCI-1588 card in the Reson 7P in order to provide the hardware-assisted implementation of PTP required for the software to operate. This card, Figure 5, provides a temperature controlled quartz crystal oscillator nominally stable to 1-2ppm and varying at 2ppm with temperature in normal ranges, and the associated ethernet hardware to implement the standard. The card implements the 'follow on' messages described above, significantly improving the synchronization performance of the system. The claimed nominal performance on a simple switched network is that the cards should synchronize to within approximately 75ns (rms) with worst-case of ±10µs [8], which is consistent with the performance found in practice during development of SWGM [6] although such extreme worst-case performance values were not observed. In order to test the performance of the integration in a form as close to that envisioned for the final implementation (Figure 3(a)) as possible, we chose to force the Reson 7P's PCI-1588 card to act as a subordinate timepiece in the PTP protocol, with the assumption that another PCI-1588 would be installed elsewhere to provide the SWGM correspondence packets and a master PTP timepiece.

The SWGM code was integrated with the Reson 7P operating software by starting the threads as the system is initialized and then having the 7P's startup thread monitor the SWGM algorithm until it indicates that synchronization has been achieved and that uncertainty estimates for the timestamps being generated are available. The call to obtain SWGM/UTC time was then placed immediately after the "Ping Now" command in the central run loop of the MBES controller, with all of the immediately surrounding section running with its software priority raised as far as possible. (This ensures minimum latency in obtaining timestamps from the hardware since it makes it less likely that the SWGM host code will be de-scheduled during the process.)

In the modified version of the 7P software, the timestamps generated from the SWGM algorithm are used to construct the timestamps for the output data set. The resolution of the data stream as currently defined is an IEEE 754 single precision floating point number for seconds, giving an effective resolution of approximately 6 s.f. in decimal notation. The resolution is a function of the magnitude of the seconds count, however, and varies up to $\sim 2^{-18}$ s, or ~3.8µs at the limit of 59.99… s. The achievable precision is therefore on the order of microseconds. Since we expected performance on this scale or better (at least in hardware), the Reson 7P software was modified to record the timestamps being observed for each ping into a separate ASCII file. In order to make performance monitoring simpler, we also modified the code to monitor edge transition events on the PFI-1 and PFI-2 inputs of the PCI-1588 card, and record full (nanosecond) resolution PTP and SWGM/UTC timestamps for these into separate ASCII files. Since we dedicate PFI-2 for ping synchronization events in this experiment (i.e., an edge transition that occurs whenever the sonar emits acoustic energy), the events are recorded into the same file as the timestamps generated by the software calls. (PFI-1 is used to monitor 1PPS events at all cards to allow for QA of the SWGM algorithm.) This is simply an expedient for efficient processing of results.

This arrangement with edge events is only required for testing; in practice, we would not need to monitor the 1PPS events at any device except the master clock, and monitoring of the pulse synchronization events is only required if the latency of reading the timestamps in software is too high, a subject to which we return in Section 5. For test, however, this arrangement allows us to investigate the quantization effects of truncating the timestamps being generated, and to provide high precision "ground truth" timestamps to quantify performance of the algorithms. This was maintained for all experiments described here except for that in Section 3.3 using the Hybrid method of Figure 6(c). Here, hardware fan-out limitations of the 1PPS output from the POS/MV meant that we were unable to maintain the 1PPS monitoring on PFI-1 of both cards, illustrating clearly the difficulties in distributing hardware 1PPS events to multiple end-points reliably.

3 Experimental Methods

All experimental work was done using a Reson 7P prepared by Reson with the National Instruments PCI-1588 card integrated with the Reson 7125 MBES topside control software. This processor was fitted to the normal survey suite of the NOAA Ship BAY HYDROGRAPHER, which consists of a Applanix POS/MV 320 V4 using BD950 GPS receivers with a baseline of 3.27m, and a standard PC workstation running Windows XP and Triton Imaging ISIS for data capture. Differential correctors for GPS were received using a Trimble DSM212L receiver; for the purposes of the testing, which took place in Solomons, MD between 2007-04-02 and 2007-04-06, correctors from the Coast Guard beacon at Annapolis, MD on 301 kHz were used. (The straight-line distance from Annapolis to Solomons is on the order of 90 km; the beacon is rated for a range of over 290 km.) In order to provide the master timepiece reference signals, and to develop the SWGM correspondence packets, the data capture computer was fitted with another PCI-1588 timing board, and the boards were connected using a LinkSys EG008W 8-port Gigabit ethernet workgroup switch and standard Cat-6 ethernet patch cords. Suitable connections from the POS/MV were made to the data capture computer as required by the SWGM algorithm and detailed below. Details of make, models and serial numbers are summarized in Table 1.

A total of four experiments were conducted as outlined below, testing the basic performance of the PCI-1588 cards, the synchronization of the integrated Reson software, and the performance of the same system using the Hybrid and Single-oscillator modes.

TABLE 1: SURVEY EQUIPMENT SUITE FOR THE TIMING EXPERIMENTS ABOARD THE NOAA SHIP BAY HYDROGRAPHER.

Device	Make	Model	Serial /Version Number
Multibeam Topside Processor	Reson	7P	51515
Multibeam Topside Software	Reson	N/A	2.13.8.2
Multibeam Sonar Head	Reson	7125 (400kHz)	5004346
Multibeam Sonar Head Firmware	Reson	N/A	MR3
Data Capture Computer	Dell	Dimension 9150	EM8FM91
Data Capture Software	Triton Imaging	ISIS/SONAR	7.0.41
Motion Sensor	Applanix	POS/MV 320 V4	2084
Motion Sensor Firmware	Applanix	N/A	2.8-6
GPS Differential Corrections Receiver	Trimble	DSM212L	-
Sound Speed Profiler	SeaBird	SBE19	19P37217-4677
Sonar Head Sound Speed	ODOM	Digibar	98376
Timing Ethernet Switch	LinkSys	EG008W	RDV004100759

3.1 PCI-1588 Card Synchronization Baseline

The survey suite detailed in Table 1 were configured as indicated in Figure 6(a), with the PCI-1588 card in the data capture computer acting as master timepiece. The multiple connections of the 1PPS signal from the POS/MV were used purely for testing purposes: the configuration ensures that each card receives a uniform event that should occur at each card simultaneously. In order to ensure this, cable lengths were matched for all branches of this circuit. The POS/MV was configured to output ZDA NMEA strings [9] to the data capture computer at 9600 baud, 8N1; the precision in this scheme comes from the 1PPS signal rather than from the ZDA string, so high speed, low latency, and minimization of traffic on the line are not required in this communication channel.

The ship systems were then started with the ship tied to the dock under shore power, and the computers were booted as for normal survey operations. The development version of the SWGM software was started on both the Reson 7P processor and the data capture computer, the machines were allowed to stabilize for 15 min. and then were allowed to run overnight without interruption or other processes running on the machines for a total of 15 hr. The SWGM driver code records the PTP and corrected UTC time of each event that occurs on the hardware inputs on each card, which are subsequently used to determine the synchronization performance of the cards. This configuration tests the baseline 'best case' performance of the cards in the ship's environment but with minimal complications of electrical/machinery noise and temperature variabilities.

During all of the other experiments described below, we recorded the 1PPS events on both master and subordinate cards, and then aggregated this data for analysis where possible. This configuration tests the baseline 'nominal' performance of the cards in the ship's environment with all of the environmental variability that typically occurs in practical applications.

3.2 Integrated SWGM/Reson 7P Synchronization Performance

In order to test the performance of the integrated SWGM algorithm, the survey suite was configured as indicated in Figure 6(b). The addition of the trigger output pulse from the Reson 7P processor allows the PCI-1588 card to determine, as well as possible, the PTP (and hence UTC) time of the outgoing sonar pulse, allowing comparison with the timestamps developed in the datagrams. There are two tests being conducted here: how well are the hardware triggers timestamped? and how well could we generate timestamps if these hardware triggers were not available?

The ship's systems were started as before, and the development version of the SWGM software was started on the data capture computer and allowed to initialize as master timepiece. The modified Reson sonar control software was then started on the Reson 7P system, and the systems were allowed to synchronize for approximately 30 min while under transit to the survey site. The ship then conducted hydrographic survey operations, conducting a patch-test (starting with an SSP measurement for calibration), and then surveying the area indicated in Figure 7. Total survey operations time was approximately 3 hr. In addition to the normal data captured via Triton ISIS, the modified Reson software recorded the PTP and corrected UTC timestamps on the hardware inputs of the PCI-1588 card, and the development SWGM software did the same for the PCI-1588 in the data capture computer. These data were preserved as structured ASCII text files to allow for sanity checking of the card synchronization, and comparison of the hardware and the software derived timestamps.

3.3 Baseline 'Standard' Timing Performance

The survey suite was kept configured as shown in Figure 6(b). The Reson 7P control software was replaced with conventional (i.e., non-SWGM enabled) software, and configured for no time synchronization with the remainder of the system components, in effect forming a Single-oscillator model.

The ship's systems were started as before, and the development version of the SWGM software was started on the data capture computer in master timepiece mode, and then on the Reson 7P computer in subordinate timepiece mode. The computers were allowed to synchronize for over 30 min., and the ship then conducted survey operations as in Section 3.2. Total survey operations time was approximately 1.5 hr. At each PCI-1588 card, the software recorded PTP and corrected UTC timestamps for all events applied to the hardware inputs, providing for analysis as before.

The survey suite was finally reconfigured as shown in Figure 6(c), the current timing configuration for the system, in effect a Hybrid Timebase system. The conventional Reson 7P control software was used, but with the Reson 7P configured to expect time synchronization using its normal operating mode of 1PPS events from the POS/MV 320 and serial ZDA messages for epoch. The development version of the SWGM software was started on the data capture computer and Reson 7P as before, and the same synchronization and survey protocols were implemented. Total survey operations time was approximately 1.5 hr.

4 Results

4.1 Basic Hardware Synchronization Performance

As collected, the timestamps at master and subordinate timepieces are stored as separate seconds and nanoseconds records (in 32-bit integers) and have nanosecond precision, but the unit seconds are defined with respect to Unix epoch (1970-01-01/00:00:00.0···) and are therefore currently (2007-04-06) in the 1.17×10^9 range. The dynamic range of these timestamps is therefore in excess of what can be represented in a double-precision IEEE 754 floating-point number (generally 15-16 s.f. in decimal notation) were we to combined the seconds and nanosecond components. In order to allow for simpler computation we reduce the dynamic range by removing a common offset from the seconds component in the master and subordinate timepiece records before we form uniform timestamps, using the minimum of the two records. This reduction allows for records up to approximately 72 hrs in length to be processed in double-precision floating-point without loosing nanosecond precision.

Since the 1PPS events by definition occur exactly on the UTC second as defined by the GPS receiver clock, the error in synchronization between the timepieces and UTC time as defined by the POS/MV outputs (in effect, latency of the hardware timestamped events) can be determined by computing the difference of the computed timestamps from the nearest integer second. The results, Figure 8, show that the performance observed in the master timepiece dominated by bursts of noise separated by areas of almost ideal performance (i.e., is 'bursty'), while that of the subordinate timepiece is more stable. The difference in performance between the master and subordinate timepieces is readily explained by the effects of scheduling on the master timepiece: on some occasions, the correspondence packet is received, processed, and stored at the master timepiece's receiver thread before the user application attempts to correct the PTP time for the event into UTC time. Since the correspondence is consequently very close to the event time, the correction is significantly better than at the subordinate timepiece where no such synchronicity is possible. The bursty performance of the master timepiece here is harder to explain, but is most likely due to intermittent thermal effects: the master timing card was installed, due to space constraints within the capture computer, next to the graphics card, which generates a significant amount of heat. A similar anomalous effect was observed in the quiescent subordinate timepiece tests, Figure 9, which clearly shows significantly increased drift variability towards the end of the run. (The drift of a timepiece is the excess time (above or below the nominal value) ticked out by the timepiece in a unit second.) This burst of data occurs 14.3 hrs into the run, and lasts for approximately 45 min. after which the behavior returns to normal. This period corresponds to the start of day activities aboard the ship, suggesting that performance might be affected by significant differences in operating environment (e.g., environmental thermal effects).

A suitable summary of the behavior of the timepieces is given by the probability density estimates for the prediction errors, Figure 10. The quiescent performance is 36 ns (rms) and 144 ns (rms) for master and subordinate timepieces, respectively; for active performance, it is 30 ns (rms) and 107 ns (rms). Again, the master timepiece results are modified by the scheduling effect outlined above; the subordinate timepiece performance is more indicative of the actual performance of the algorithm. The small difference between these two tests is most likely due to statistical fluctuation of the estimates (i.e., standard error) and the difference in run-time of the tests, rather than any true difference in the behavior of the algorithm while quiescent and active.

The long-term stability of the oscillators and their driving noise performance can be determined from the Allan deviation [10], a common measure of oscillator stability. In principle, the Allan deviation describes the expected difference between true and indicated time over specified time duration, averaged for the time between observations:

$$\sigma_y^2(\tau) = \frac{1}{2(N-2)\tau^2} \sum_{n=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \quad (4)$$

(where there are N observations of time, x_k, at intervals of τs.) which is a measure of the likely drift of the timepiece's indicated time over the given duration and hence of the oscillator's stability. Figure 11 shows the performance of the timepieces in quiescent and active modes, based on the PTP timestamps for the events, and the corrected UTC times. The base performance of the hardware oscillators are indicated by the PTP times, which clearly shows the long-term drift that is expected from the cards. (Note that the Allan deviation shows drift averaged over the observation period; similar deviations over longer periods therefore imply larger absolute drift.) Of course, the subordinate timepiece follows the master timepiece, and therefore the Allan deviation is a little higher. The Allan deviation of the corrected UTC times shows significantly improved performance with lower drift. In effect, the timepieces now follow the 1PPS signals from the POS/MV exactly. This does not imply that the timepieces are now perfect, just that they follow the 1PPS signals without significant drift over the observation period.

4.2 Real-time Data Synchronization Performance

After confirming the base performance of the timepieces as in Section 4.1, the timestamps recorded in the XTF data archive were extracted using custom software, and transformed into a uniform timestamp representing nanoseconds since Unix epoch (1970-01-01/00:00:00). For the SWGM-enabled configuration, Figure 6(b), the ASCII debug timestamp files were used; for the Hybrid configuration, Figure 6(c), the timestamps used were those in the Reson datagrams encapsulated in the XTF; for the Single-oscillator configuration, the timestamps recorded in the XTF file headers were used. (The dynamic range required for this cannot fit into a 32-bit integer; 64-bit computations were used throughout.) The separately recorded trigger timestamps were synchronized to the datagram timestamps and direct differences between corresponding timestamps were computed. The mean offset in the trigger pulses was computed and subtracted in each case; there is an intentional delay in the Reson 7P hardware which is nominally 5 ms.

Statistics of the SWGM-enabled 7P processor are shown in Figure 12. Since we do not have a better reference timepiece, the only way to determine the performance of the hardware timestamps is to use the estimated hardware timestamp uncertainty, which in this case is equivalent to the latency. This has previously been shown [6] to match well the observed true error in timestamps in this configuration. The results, Figure 12(a) suggest that the hardware timestamps are on the order of at worst a few hundred nanoseconds (100-300ns (rms) with a modal value of approximately 150ns (rms)). This is slightly higher than the 1PPS event timestamps of Figure 10 since the ping times are distributed randomly with respect to the synchronization events. The nominal latency we would observe if all that was available was a call in software to the hardware timestamp unit for the event time is on the microsecond scale, with standard deviation of 28 µs (rms), Figure 12(b). (We note that there are occasional outliers in this sequence as indicated in Figure 12(b), which are the subject of current study; see Section 5.) The asymmetric distribution of timestamp latency is likely a function of the differing ping repetition frequencies, Figure 12(c), since the internal loading of the 7P processor is a function of the ping frequency. This increase in latency is due to the inherent problems of obtaining a timestamp at the user software level: the exigencies of operating a time-shared (pre-emptive multitasking) computer means that there may be a significant time lag between requesting a timestamp and receiving it. Since it is not clear at which point in those events the timestamp represents, the uncertainty of the timestamps, Figure 12(d), can also be very high, even with faster, modern computers (the Reson 7P used in these experiments is a dual Xeon machine, for example). We return to the difficulties of software-based timestamps in Section 5.

The pdf estimates of the timestamp differences for the Hybrid and Single-Oscillator models, Figure 13, show that there are significant differences between the three configurations. The standard deviations of the differences are 288 µs (rms) and 13.6 ms (rms) for the Hybrid and Single-Oscillator models, respectively. (In the case of the Single-Oscillator model, the mean latency has been removed, so that this is an estimate of latency variability, rather than absolute latency.) Clearly, there is a significant advantage in using the SWGM enabled system, although the shape of the Hybrid mode pdf, Figure 13(a), suggests that the majority of the latency observed is in fact coming from the truncation to millisecond precision of the timestamps due to the limitations of the timekeeping hardware used in this case (i.e., the Windows system clock with an auxiliary hardware millisecond counter). The actual synchronization performance of this system is therefore most likely better than observed although this is not evident to the user, a problem to which we return in Section 5 below. The uniform nature of the error in this case implies that we should really cite the likely latency as ±0.5ms rather than as a standard deviation.

The standard deviations of the latencies were then computed in windows over all of the observed data in each example, Figure 14. In the case of the SWGM-enabled system where there is a contiguous time-sequence of data, these are 10 min. windows; for the other experiments, these were computed on each coherent segment of ping data. The differences observed can be explained by the nature of the distribution of the timebase in each case, and the latency of the data transmission as a function of time. Again, the effect is smallest in the SWGM-enabled system since the latency is very low and as stable as the timing ethernet's performance, and higher in the Single-oscillator model. This can have significant implications in applications; for the purposes of uncertainty modeling, for example, the worst-case latency would have to be assumed so as to avoid under-estimation of the associated uncertainty.

5 Discussion

The results demonstrated in these experiments show that the method of time synchronization chosen can have a very significant effect on both the mean latency in the data, and on its variability. The results of Figure 10, in particular, show that it is possible, with suitable PTP/SWGM enabled software, to reduce the latency in any hardware timestamped system to a sufficiently low value as to be considered essentially zero for any surface hydrographic system: 500ns latency at 15kts is equivalent to an offset along-track of approximately 0.004mm[1]; on a dynamics signal of magnitude 10° and period 1Hz, 500ns latency is equivalent to a maximum angular uncertainty of approximately 0.0000314°. Our experience with these cards is that hardware timestamps with effective latency of ~100-150ns (rms) is more common.

The Hybrid model appears to give performance at worst ±0.5ms, due to the quantization of the timestamps to 1ms. It is, unfortunately, impossible to improve on this quantization without incurring equivalent complexity to a scheme such as the PTP/SWGM one proposed here (in which case it is almost surely simpler just to source extant cards rather than redevelop an equivalent). It is not readily apparent what the actual synchronization of the system is, although we expect it to be significantly better than that observed. In practice, however, since the highest resolution reliable timer is only at millisecond precision, it hardly matters: what the user observes is an uncertainty uniformly distributed over ±0.5ms. This is for many purposes, however, quite adequate. With the same (extreme) test conditions above, this corresponds to 4mm along-track and 0.0314° angular offset, both of which are approaching the noise limits of the appropriate sensors in most cases.

The single-oscillator model is clearly inadequate for modern survey systems, both because of the magnitude and the variability of the latency. This scheme should now be considered obsolete for all practical purposes of hydrographic survey.

We have not tested here the latency behavior of the Distributed Serial Message model of Figure 1(c); the survey system configuration used did not support it. It is likely, however, that since it relies on the stability of transmission of the UTC message from the POS/MV (or other timebase source) on a high-speed serial line without the 1PPS synchronization or millisecond timer of the Hybrid model, it is likely that the performance is of the same magnitude as, but a little worse than, the Hybrid model examined here. In particular, we might expect that the variability of latency would be higher due to serial buffer effects. It would be straightforward to apply the same methods used here to this problem, however, if more precise information were required.

The results of Figure 12 emphasize for us that in the case where hardware event timestamps cannot be generated[2], the actual latency performance of the system might be more influenced by task priority, loading and scheduling effects within the host Operating System than the quality of the hardware time synchronization. (This might occur, for example, in timestamping data from "dumb" sensors, such as cable counters, where the only available indication of data arriving is a serial message.) In essence the problem

---

[1] For reference, the diameter of a human hair is typically in the range 0.017-0.180mm.

[2] It is possible in some IMU systems, such as the one here, to have timestamps generated corresponding to some hardware input. These might be used as an alternative to the sort of timing cards described here although this method is not universal, the number of inputs is limited and proximity of equipment (or lack therefore) might induce more cabling than would be desired in this case.

is that the software thread requesting the timestamp might be descheduled during the attempt to obtain the timestamp from the local hardware, leading to an essentially unknown potential for latency. We estimate the actual latency in the SWGM algorithm, but this only quantifies the problem. Some mitigation techniques can be used, however, if this is a significant problem. Previous experiments [6] have shown that raising the priority level of the thread making the timestamps reduces the probability of being descheduled during the timestamp generation process; ensuring that the timestamp generation is as close as possible to the data creation event avoids as many schedule instants as possible; minimizing any calls for system resources (e.g., disc I/O) similarly avoids scheduling events; and using faster processors generally helps. To a certain extent, this is simply a penalty that is enforced for the benefit of having a prioritized, pre-emptive, multi-tasking Operating System, and is essentially unavoidable at some level. In the future, better integration with the timestamp generation hardware might improve this situation somewhat by quantifying better at which point in the call to obtain the timestamp the returned time actually applies. In the meantime, however, software-derived timestamps, even from hardware synchronized timepieces, will likely remain limited to the microsecond level.

Our test integration of PTP/SWGM with the Reson 7P processor control software appears to work essentially as expected. We do, however, observe occasional single point outliers of significantly higher latency in software-derived timestamps than we would expect. These are evident as tails of the distributions in Figure 12. There are at least two potential explanations for these. Firstly, we observed during the experiment that the ping trigger pulse generating the reference hardware timestamps suffered from occasional noise effects (we suspect from a damage coaxial cable). These have been removed from the record as much as possible during the post-processing, but it is possible that either the noise affected the timestamps in an unpredictable way on some occasions, resulting in the outliers, or that the post-processing corrections did not adequately account for all of the effects observed. Secondly, there may be occasional latency effects due to scheduling or data loading within the 7P processor. We are currently investigating these effects, although we note that they should not in practice affect a correctly working hardware timestamping system.

We did not pay particular attention to environmental factors during this experiment, although the primarily limitation on oscillator stability is typically temperature. This was done intentionally to gain an idea of the likely real-world performance of the systems. Direct observation of ambient temperature in the survey cabin varied from 25°C to 30°C depending on the outside air temperature and functionality of the interior air conditioning; no significantly different behavior was observed, suggesting that local thermal environment (i.e., within the host PC) is more significant than ambient conditions. This also appears the most likely cause of the bursty error performance of the master timepiece, Figure 8, which is unlike the behavior seen in other tests with identical timing hardware and software. Our working hypothesis for this is that proximity to the graphics card in the data capture machine, and the higher ambient temperatures inside the case due to a faster CPU than in our previous tests, resulted in bursts of thermal disruption. The effects are small and therefore probably not sufficient impetus to take remediating action; if the very highest precision was required, however, some form of auxiliary cooling or thermal stabilization might be required, depending on local hardware.

We have considered here essentially ideal conditions for PTP/SWGM to operate: a separate, unloaded switch for the timing traffic and sufficient time to synchronize before starting the experiments. Relaxing these restrictions would likely have a deleterious effect on performance. The essence of PTP is in predicting the network latency between master and subordinate timepieces (Section 2.1); adding extra traffic to the network switch hosting the timebase distribution increases the potential for this latency to change more rapidly, which would be reflected in higher data latency variability (the mean latency would still be zero in the long term). At present, however, most PTP implementations are still 100bT ethernet based, and therefore would not likely have sufficient bandwidth for current generation high resolution hydrographic systems. This makes it likely that the PTP cards would carry primarily timing traffic along with some auxiliary information (e.g., low density command and control data) for the immediate future, circumscribing the problem.

The time required for adequate synchronization of PTP cards is not well established; the 15-30 min used here was intentionally conservative and most likely vastly in excess of the minimum required. This time is required to allow subordinate timepieces to slew their internal time representation to within an acceptable level of the master, and therefore depends on the initial offset between the timepieces, the rate of synchronization messages and the slew rate of the oscillators. We have anecdotally observed this taking up to a minute to occur in some cases with our current SWGM implementation, although this could likely be improved by some additional coding to start the PTP oscillator at the subordinate timepiece closer to the master by default. Some more experimentation is probably required to determine the effective capture time, however. At present, though, this can at least be monitored by considering the predicted uncertainty associated with each timestamp: once the uncertainty drops below the (user) required threshold, data capture can commence.

There are a number of future directions for this research. For example, the uncertainties developed by the SWGM algorithm can be used to feed Total Propagated Uncertainty models for data [11], removing another unknown uncertainty from the equations. In fact, the mean latency of all PTP/SWGM systems being zero also reduces the complexity of patch-test procedures [12] since no latency calibrations are then required. One exciting prospect is to have full PTP capability incorporated into a GPS-aided IMU so that the system runs natively over the ethernet without the SWGM software layer; combined with local hardware timestamps, this has the potential to radically simplify (and improve) timekeeping in survey systems. Finally, in this experiment we have used COTS PTP cards, but there are now embeddable [13] and IP-core [14] versions of the algorithm that can be integrated into any piece of equipment. This opens the possibility for routinely constructing PTP-enabled instruments, a very exciting prospect.

6 Conclusions

We have shown that under typical survey conditions, the proposed PTP/SWGM time distribution scheme can synchronize multiple timepieces to UTC time derived from a GPS receiver with error on the order of 150ns (rms) and therefore can reliably timestamp hardware signals (e.g., a ping trigger) at any participating timepiece with zero mean latency and latency variability on the order of 100-300ns (rms) with modal variability of ~150ns (rms). If hardware timestamps are not available, the latency variability expected increases due to scheduling, priority and loading within the host Operating System, and can be up to ~28µs (rms) in the long term, although it is more likely in the 10-15µs (rms) range instantaneously.

We have also shown that timestamps to within ±0.5ms are possible with a hybrid hardware/software scheme used on current generating Reson 7P processors, limited primarily by quantization of timestamps to 1ms levels; the true latency variability is indeterminate (and essentially irrelevant since it cannot be observed at the user level) but likely lower than this. Conventional single-oscillator timestamps were shown to have latency variabilities of up to 14ms (rms) and should therefore be considered inadequate for modern high resolution survey systems.

The proposed system, in addition to significantly better performance than contemporary equivalents, greatly reduces the complexity of integration of the timekeeping components with the survey system. In the ideal case only an ethernet connection is required. PTP systems that are embeddable (e.g., within ethernet chipsets) or integrable (e.g., as VHDL cores) are readily available; COTS ethernet cards with PTP support are also available from a number of manufacturers, making this a very attractive solution for high precision and high accuracy timekeeping within hydrographic survey systems.

Acknowledgements

The authors would like to thank NOAA (NMAO and HSTP) for allowing us the use of the BAY HYDROGRAPHER to conduct these experiments, LT/JG Michael Davidson, NOAA (OIC BAY HYDROGRAPHER) and Steve Brodet (HSTP) for their skilled, enthusiastic and diligent assistance in running the ship, and Michael Mutschler (Reson) for his dedicated, talented and sustained effort configuring and running the equipment during the tests.

Reference

[1] D. Lockhart, 2005, Personal Communication.

[2] NOAA Hydrographic Surveys Division, NOAA/OCS Field Procedures Manual, 2nd Ed., National Oceanic and Atmospheric Administration, 2006.

[3] Timing Committee, Telecommunications and Timing Group, *IRIG Serial Time Code Formats*, 200th Ed., Range Commander's Council, 1998 [online: `http://www.irigb.com`].

[4] J. E. Miller, J. S. Fergusson, J. S. Byrne and W. S. Simmons, "Shallow Water Multibeam Hydrography to IHO Standards", Sea Tech., 37(6): 81-86, 1996.

[5] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Instrumentation and Measurement Society, 2002, ISBN: 0-7381-3370-1.

[6] B. R. Calder and A. McLeod, "Ultra-Precise Absolute Time Synchronization for Distributed Acquisition Systems", Submitted to IEEE J. Oceanic Eng., 2006-09-25.

[7] K. Correll, N. Barendt and M. Branicky, "Design Considerations for Software Only Implementations of the IEEE-1588 Precision Time Protocol", in *Proc. IEEE 1588 Conference*, Zurich: National Institute of Standards and Technology, October 2005. [online: `http://ieee1588.nist.gov`].

[8] *National Instruments PCI-1588 User Manual*, National Instruments, 2005. [online: `http://www.ni.com/pdf/manuals/371670a.pdf`].

[9] *NMEA 0183 Standard for Interfacing Marine Electronic Devices*, 3rd Ed., National Marine Electronics Association, 2002.

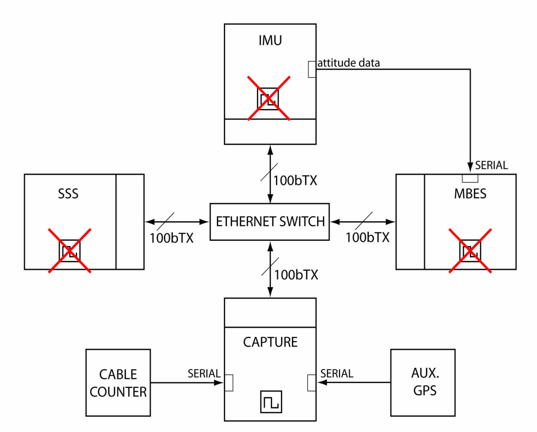
[10] D. W. Allan, "Time and frequency (time-domain) characterization, estimation and prediction of precision clocks and oscillators", *IEEE Trans. Ultrasonics, Ferroelectrics and Frequency Control*, 34(6):647-654, 1987.

[11] Hare, R., A. Godin and L. A. Mayer, "Accuracy Estimation of Canadian Swath (Multibeam) and Sweep (Multitransducer) Sounding Systems", Canadian Hydrographic Service, 1995.

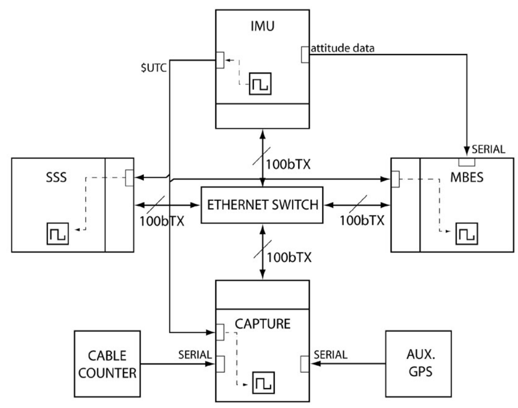
[12] Godin, A., "The Calibration of Shallow Water Multibeam Echo-sounding Systems", MS Thesis, University of New Brunswick, 1997.

[13] *Hardware-Assisted IEEE 1588 Implementation in the Intel IXP46X Product Line*, Intel Corporation, 2005. [online: `http://download.intel.com/design/network/papers/30506801.pdf`]

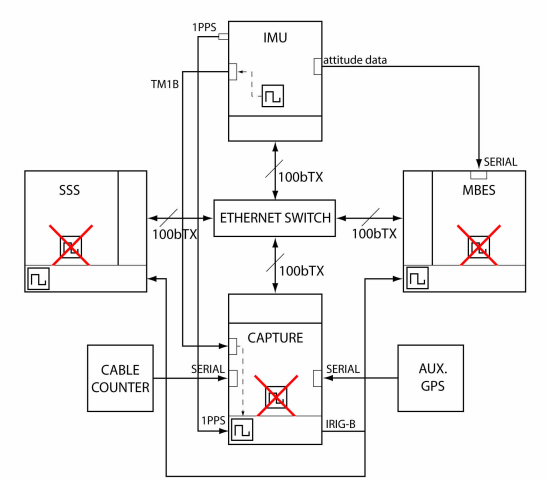
[14] *A 1588 Tri-Speed Ethernet MAC Core* (Product Brief), MorethanIP, Muenchner Strasse 199, D-85757 Karlsfeld, Germany [online: `http://www.morethanip.com/ieee_10_100_1000.htm`]



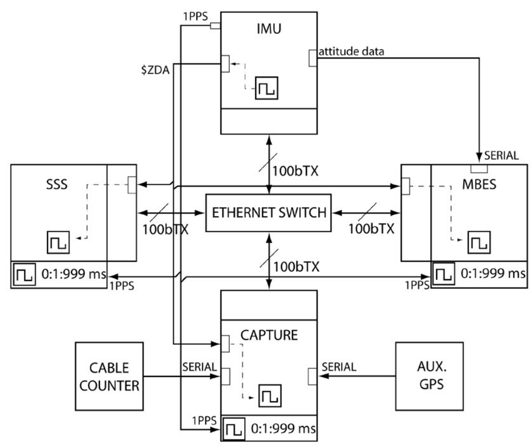
(a) Single-Oscillator Model



(b) Distributed Serial Message Model



(c) Distributed Timebase Model



(d) Hybrid Model

Figure 1: Models of timebase distribution; red crosses indicate oscillators that are not used in the timing chain for the particular configuration. The Single-Oscillator model, (a), sends all data to the capture computer for timestamping, but suffers from transmission latency issues; the Distributed Serial Message model, (b), avoids transmission latency by timestamping all data locally, but suffers from latency in distribution of the timestamps and does not scale well; the Distributed Timebase model, (c), avoids latency and has high precision due to the additional hardware components, but can have significant complexity and cost in wiring and scaling; the Hybrid model, (d), is a half-way house between Distributed Serial Message and Distributed Timebase models which can achieve high precision, but is difficult to scale.

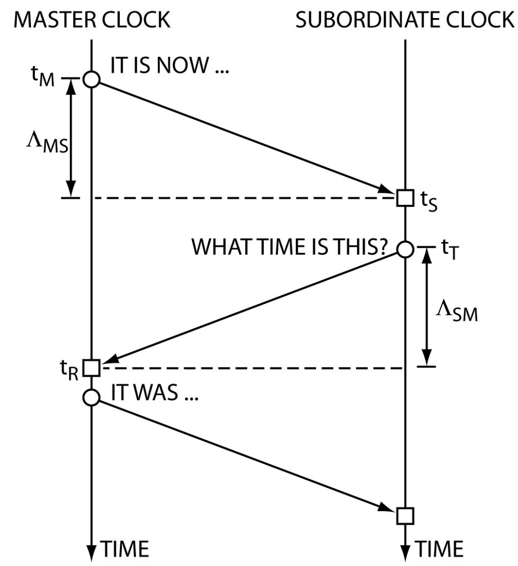
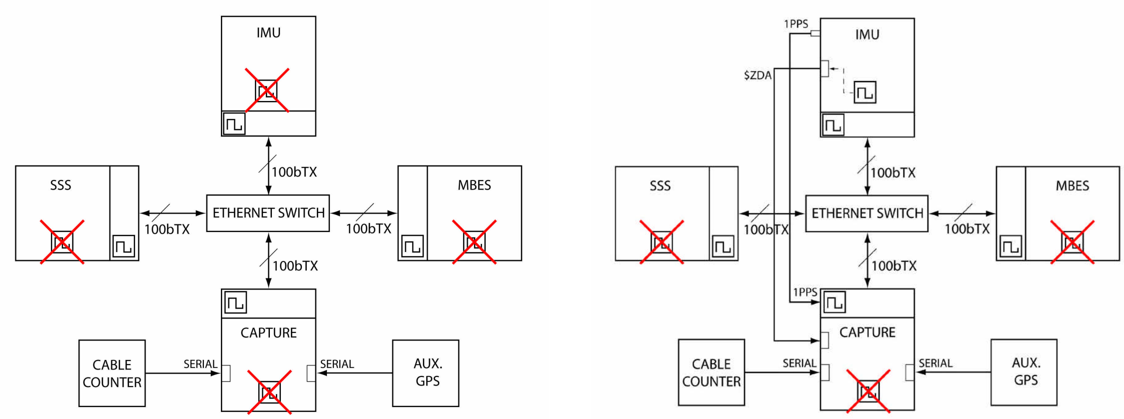


Figure 2: Generation of timestamp offset errors and latency estimates in the PTP algorithm. Exchange of high-precision timestamps derived from the PTP local oscillators within the messages exchanged over the ethernet by two timepieces allows subordinate clocks to determine their offset from the master and hence correct for it in a conventional control loop.



(a) 'Pure' PTP timebase distribution from IMU's GPS receiver clocks

(b) Software Grandmaster timebase distribution

Figure 3: Timebase distribution models for PTP in a hydrographic survey system. The 'pure' model, (a), distributes UTC time from the IMU's PTP timepiece, which is disciplined to the IMU's GPS receiver clocks. Until IMUs have this facility, a modified Distributed Timebase Model is used with PTP aiding to distribute the timebases, and Software Grandmaster corrections to provide UTC timestamps at all component systems.

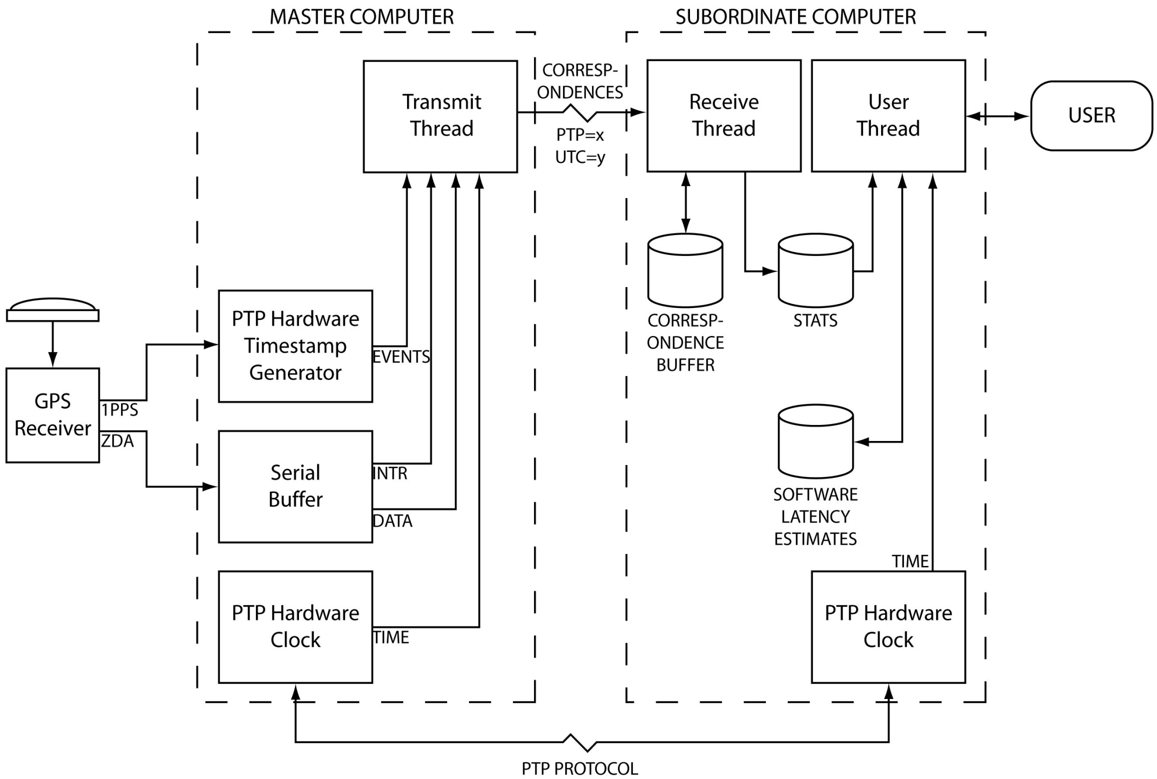


Figure 4: Outline of the Software Grandmaster algorithm. The master timepiece generates correspondences between UTC time and the values read on the PTP oscillators, and distributes these to all subordinate timepieces using the same ethernet protocols as PTP. Buffering and statistics generation at the subordinate computer allows for transparent generation of high-precision UTC timestamps at the user level.

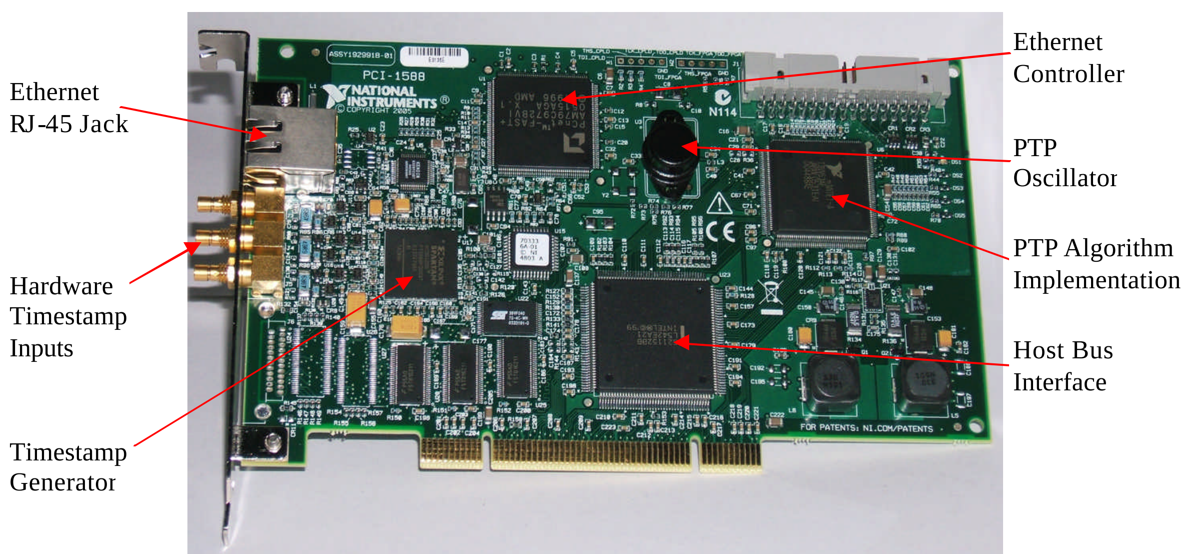
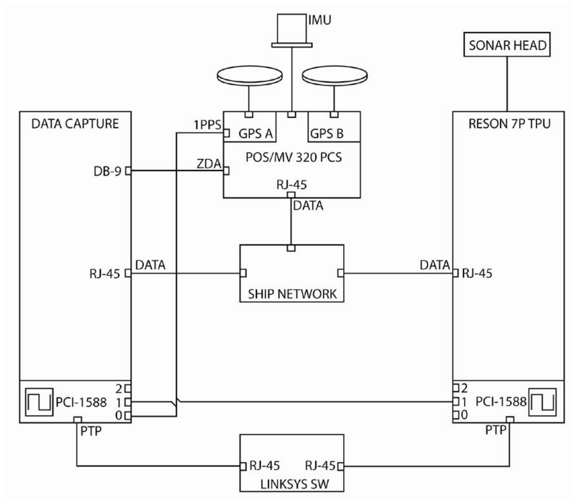
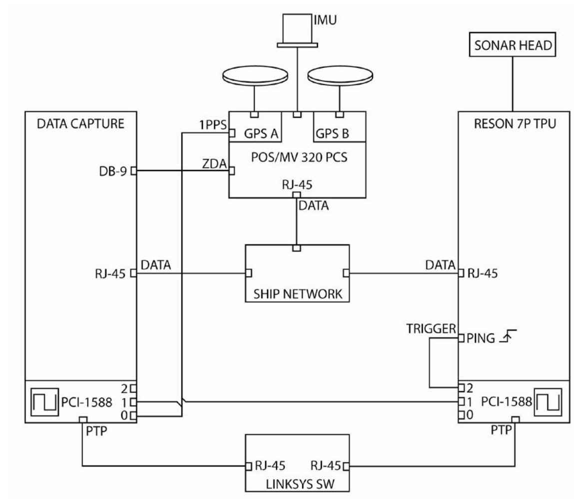


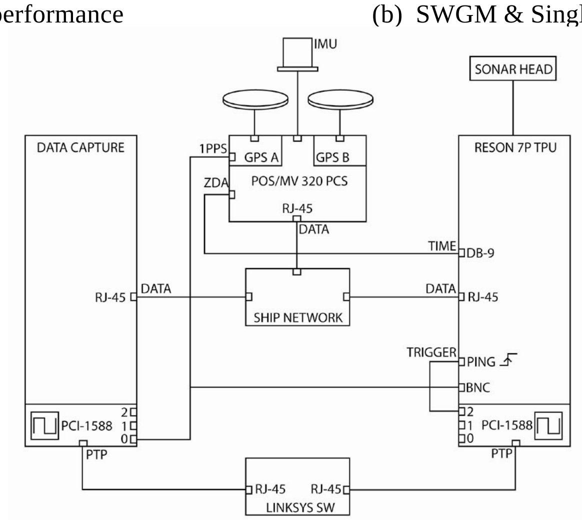
Figure 5: The National Instruments PCI-1588 card, implementing IEEE 1588-2002. The gold colored inputs at left provide edge-triggered hardware timestamps with the full precision of the hardware oscillator, ensuring high-precision synchronization. The oscillator is a temperature controlled quartz crystal oscillator nominally accurate to 1-2ppm.



(a) Baseline performance



(b) SWGM & Single-Oscillator Modes



(c) Hybrid Mode

Figure 6: Survey suite configurations for the experiments: (a) Baseline performance of the PCI-1588 cards; (b) Nominal performance of the integrated Reson/SWGM software, and Baseline performance in Single-oscillator mode; and (c) Baseline performance in Hybrid mode.

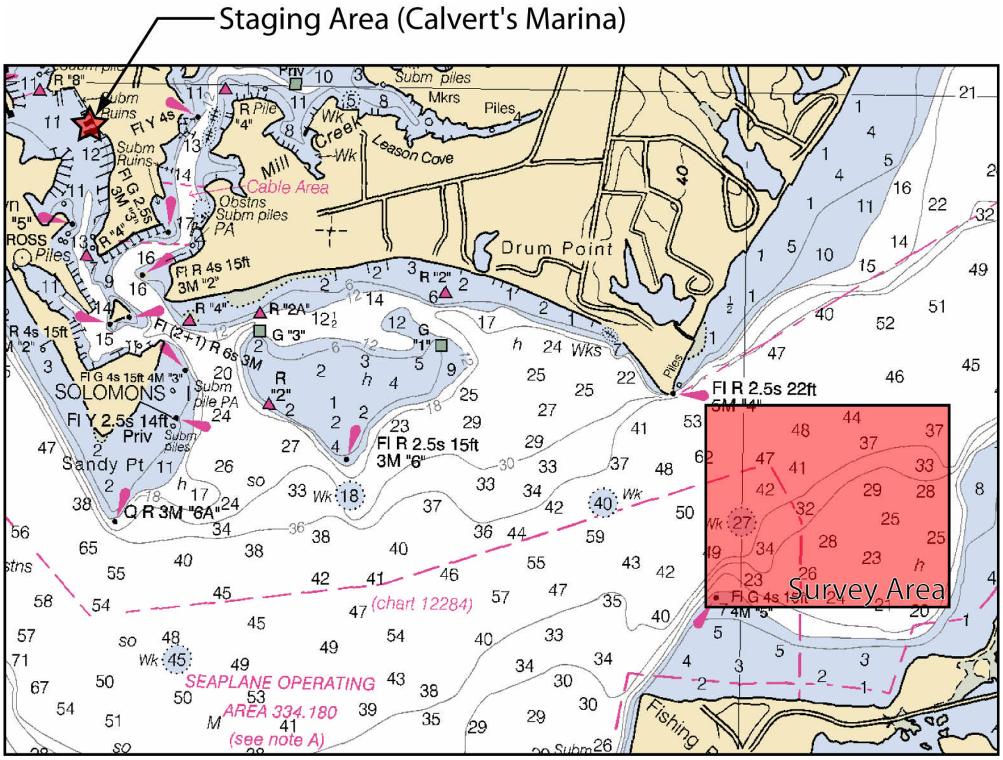
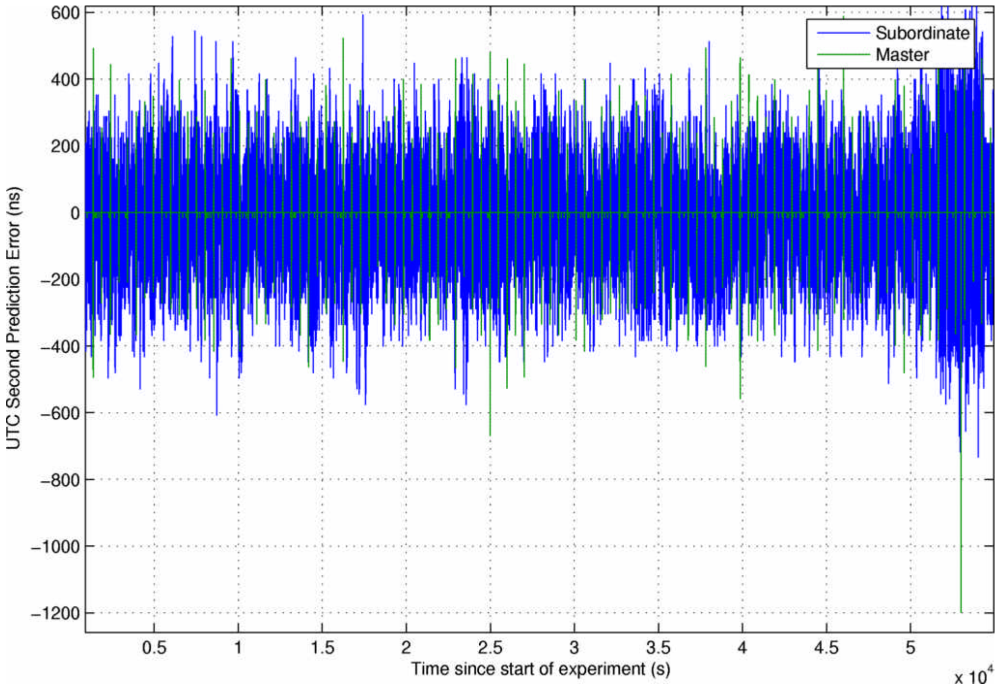
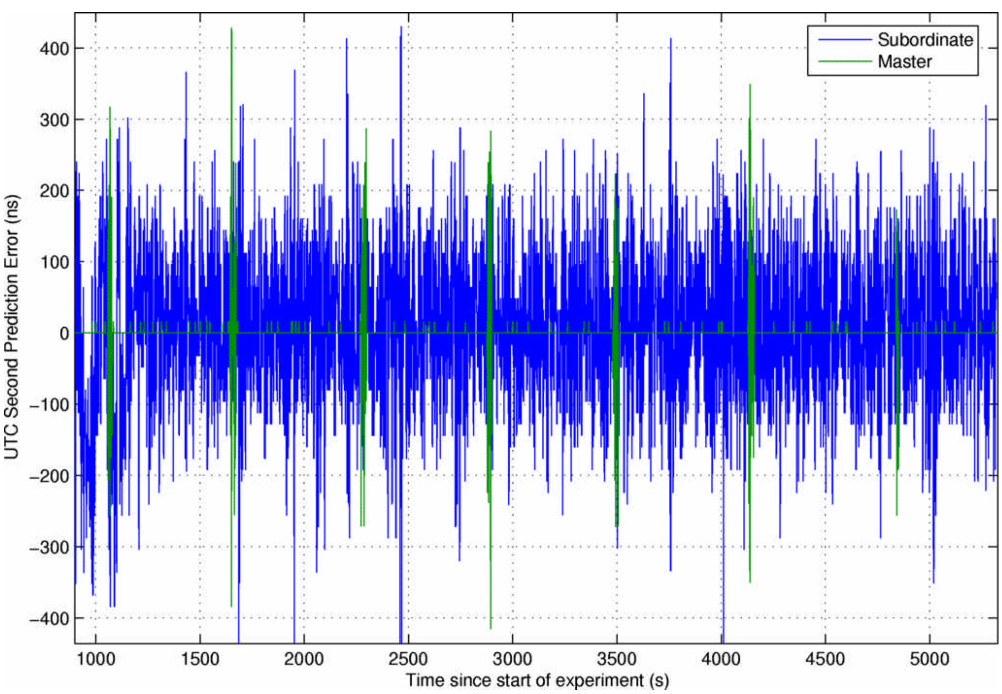


Figure 7: Location chartlet for the experiments (extracted from NOAA Chart 12684). The BAY HYDROGRAPHER was staged out of Calvert's Marine in Solomons, MD, with all on-water testing in the indicated area. This contains areas suitable for patch-test procedures, and a relatively flat area with pre-deployed man-made features for small-scale test surveys.



(a) Quiescent Testing



(b) Active Testing

Figure 8: Errors in prediction of UTC seconds events at master and subordinate timepieces for quiescent (a) and active (b) tests. The master timepiece events evidence spikes in performance, probably due to thermal shock within the capture computer; the subordinate timepiece events appear more stable because any such effects are masked by the higher overall uncertainty (due to jitter in transfer of the timebase). Absolute errors are, however, still at the nanosecond level on average.

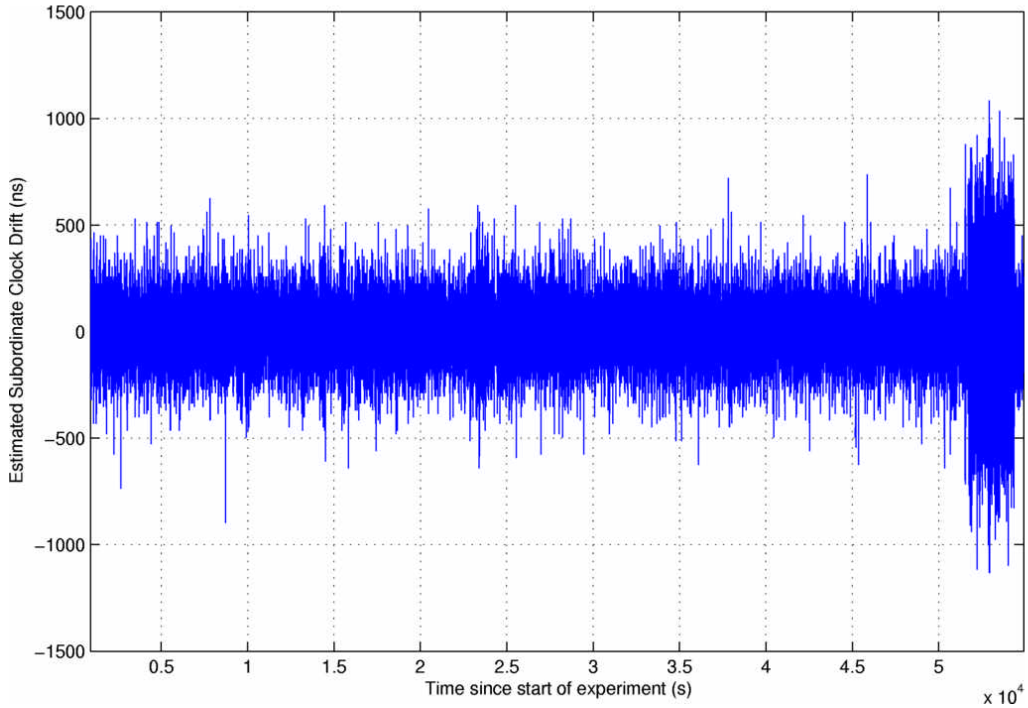


Figure 9: Time series for subordinate timepiece drift estimates during the quiescent tests of Section 3.1. The anomalously increased drift towards the end of the sequence lasts for approximately 45 min. and is apparently correlated with start of day activities aboard the test platform.

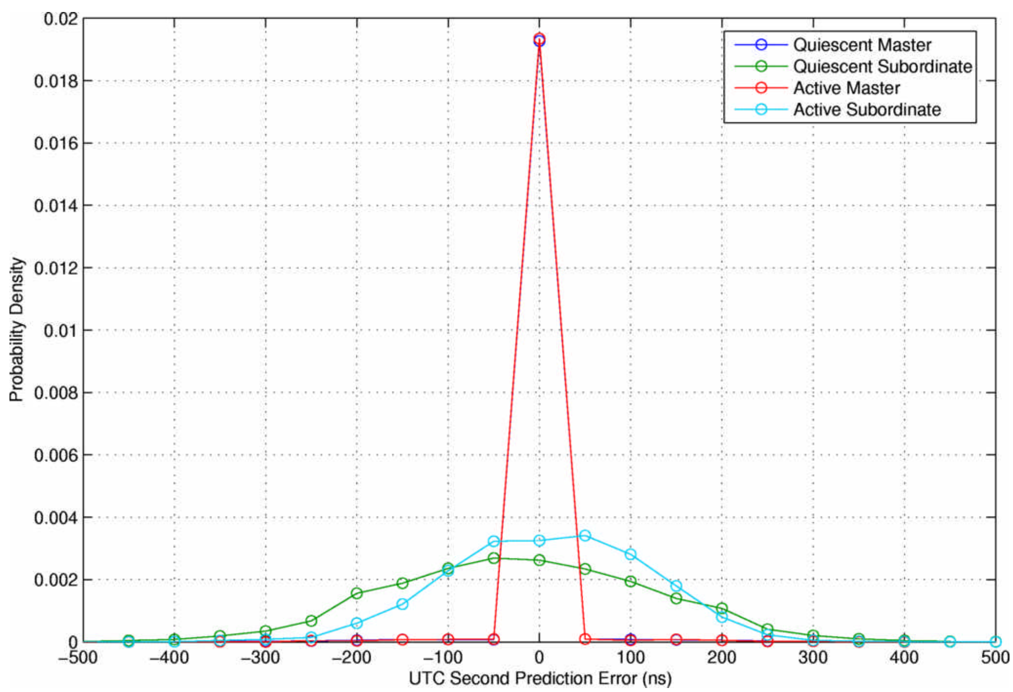
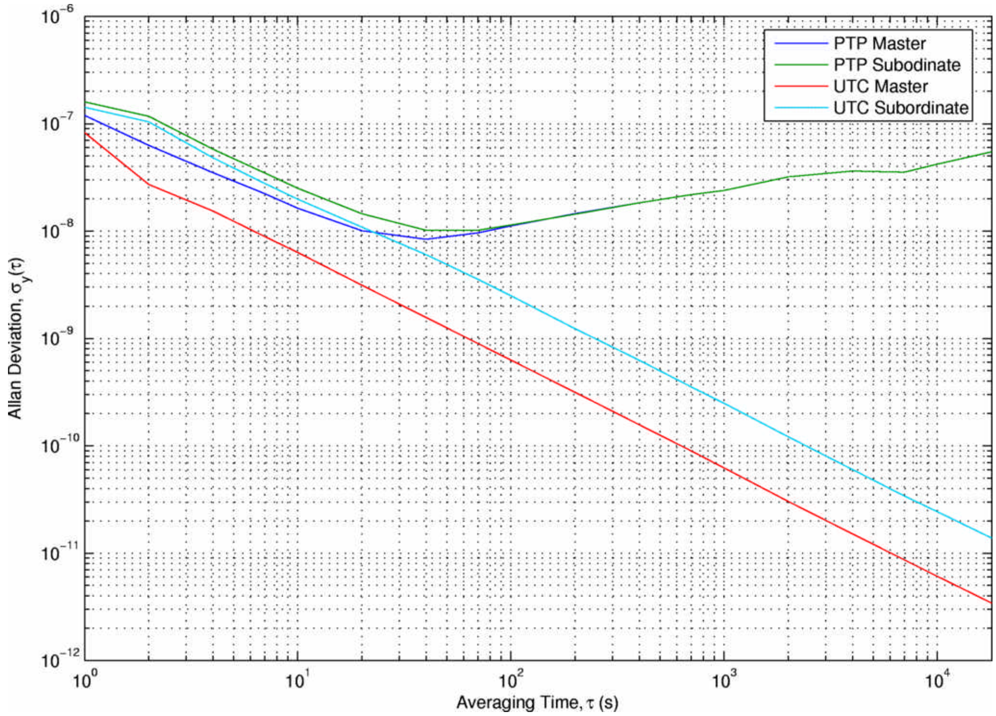
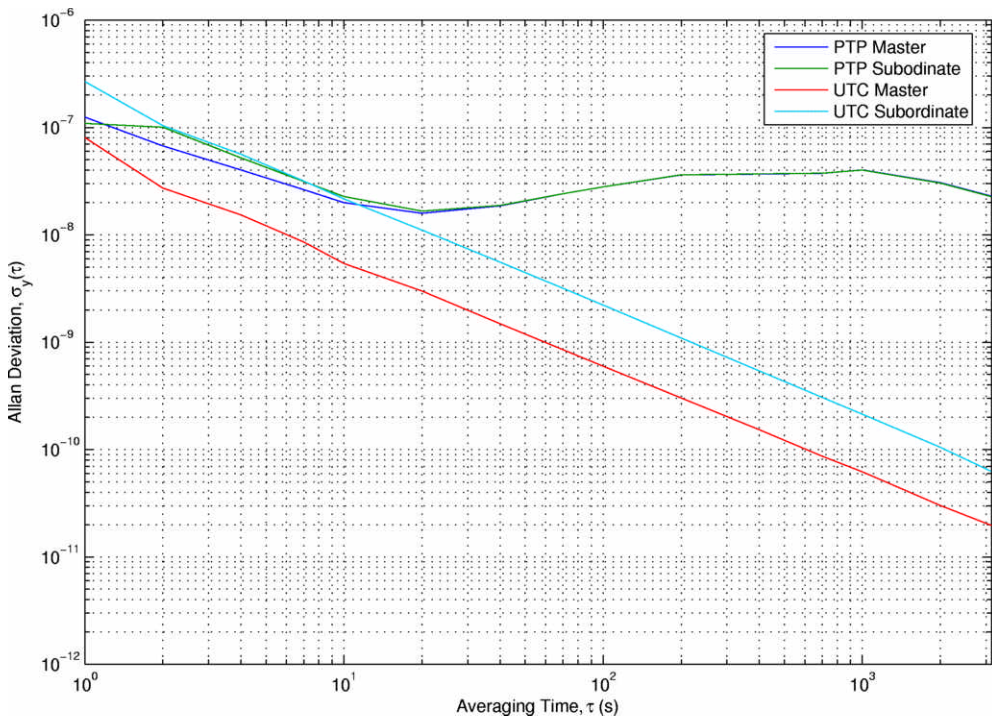


Figure 10: Probability density estimates for the observed error in UTC seconds prediction for master and subordinate timepieces in quiescent and active tests. The master timepiece behavior appears to be significantly better due to the local scheduling effect; the subordinate timepiece behavior is more indicative of the actual performance of the timing scheme.

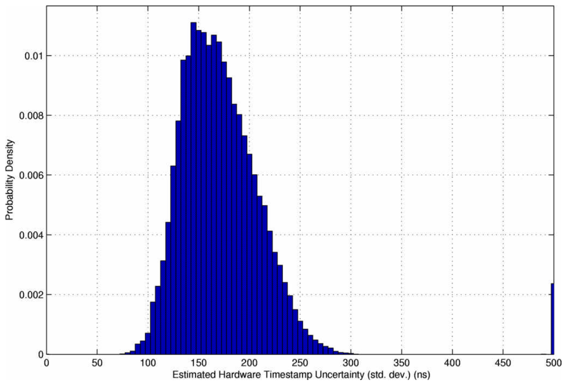


(a) Quiescent Testing

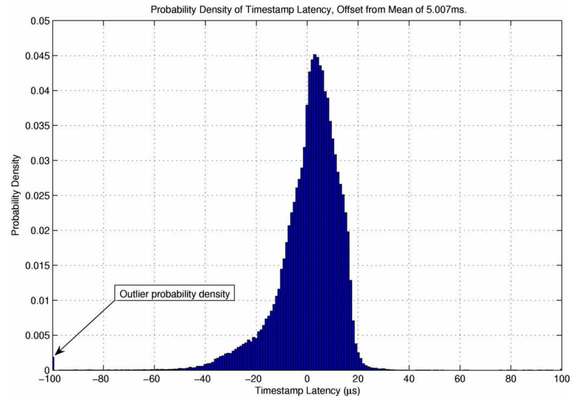


(b) Active Testing

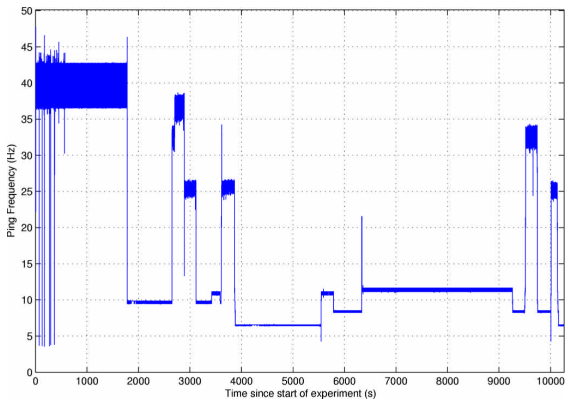
Figure 11: Allan deviation estimates for the quiescent and active tests, showing the long-term stability of the timepieces. The PTP performance is the base performance of the hardware oscillators in the PCI-1588 cards; the UTC performance shows that the long-term drift of the oscillators has been effectively cancelled by the SWGM so that the distributed timebase tracks the POS/MV's timebase without drift over the observation period.



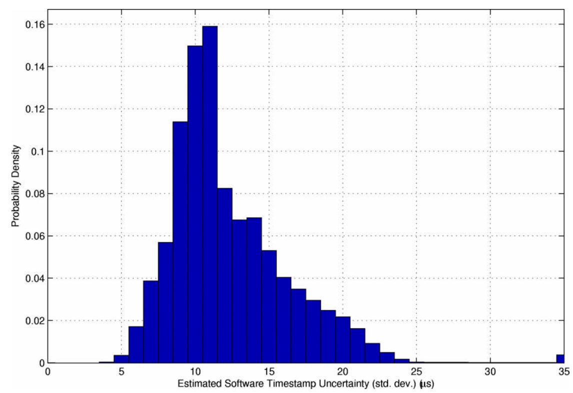
(a) Estimated Hardware Latency PDF



(b) Observed Software Latency

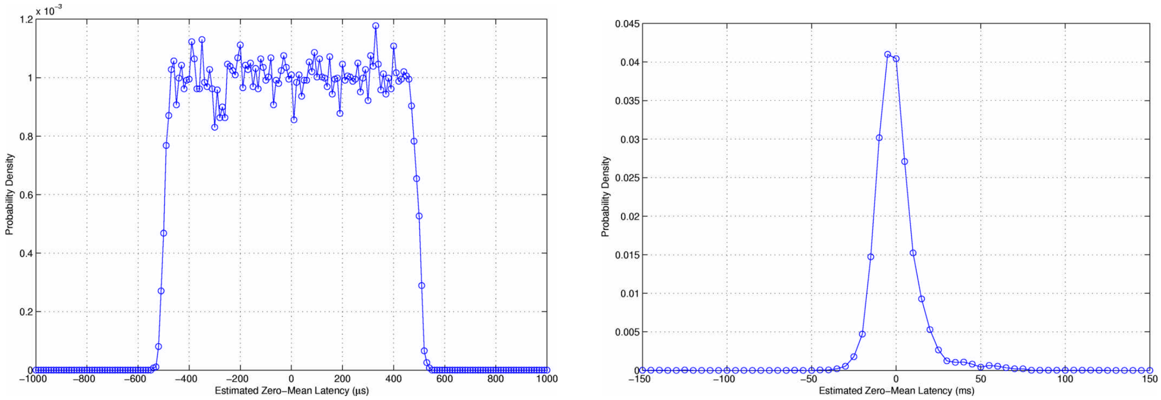


(c) Ping Frequency



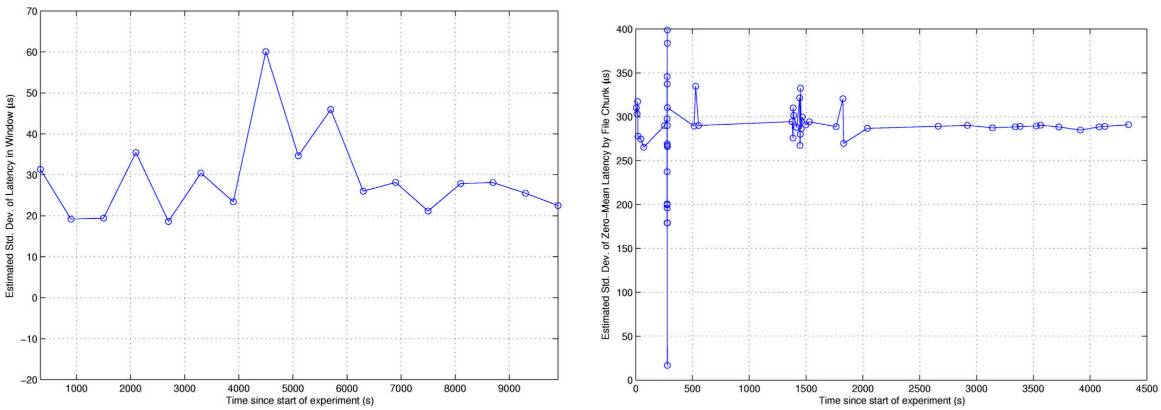
(d) Estimated Software Latency PDF

Figure 12: Active performance for the SWGM -enabled Reson 7P Processor. The hardware timestamp latency, (a), is in the range 100-300ns (rms) with modal value of approximately 150ns (rms), corresponding to the performance of the underlying SWGM algorithm in distributing the timebase. The observed software latency, however, is approximately 28 μs (rms), (b), with asymmetry probably due to differences in latency at different ping frequencies, (c); the magnitude with respect to the hardware values implies that the higher observed latency is due to scheduling and priority issues within the 7P processor; the uncertainties predicted for the software timestamps, (d), match the latency observed in practice, and indicate that although the long-term performance might be higher, the instantaneous uncertainty is likely somewhat smaller.

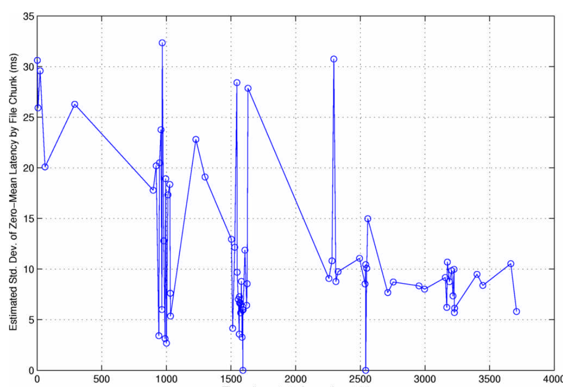


(b) Hybrid Mode (b) Single-Oscillator Mode

Figure 13: Estimates of latency pdf for the Hybrid, (a), and Single-Oscillator, (b), models. The scale of latency is significantly higher for the Single-Oscillator model than for the Hybrid model, as expected, and both are significantly higher than the SWGM-enabled Reson 7P Processor, as expected. The shape of the Hybrid mode pdf suggests that the majority of the latency is quantization noise, rather than true latency.



(a) SWGM-enabled Software Mode (b) Hybrid Mode



(c) Single-Oscillator Mode

Figure 14: Variability of latency standard deviation across experiments. The variability of latency can change considerably as a function of time, leading to difficulties with correction.