AFRL-HE-AZ-TR-2002-0066

# UNITED STATES AIR FORCE RESEARCH LABORATORY

## AN EMBEDDED SIMULATOR TEST EVALUATION MONITOR (ESTEEM) TO IMPROVE DISTRIBUTED MISSION TRAINING

**Ronald B. Ewart**
**Gary J. Slutz**

**Protobox LLC**
**1464 North Broad Street**
**Fairborn, Ohio  45324**

**September 2002**



**AIR FORCE MATERIEL COMMAND**
**AIR FORCE RESEARCH LABORATORY**
**Human Effectiveness Directorate**
**Warfighter Training Research Division**
**6030 South Kent Street**
**Mesa AZ  85212-6061**

# NOTICES

**This research was conducted under the Small Business Innovation Research (SBIR) program as a Phase I effort. Publication of this report does not constitute approval or disapproval of the ideas or findings. It is published in the interest of STINFO exchange.**

**The US Government does not require SBIR Phase I contractors to adhere to any particular format or style. In accordance with SBIR guidelines for Phase I efforts, the contractor's report is accepted for publication but is not edited.**

**Using Government drawings, specifications, or other data included in this document for any purpose other than Government-related procurement does not in any way obligate the US Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation, or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.**

**This report has been reviewed by Public Affairs and is releasable to the National Technical Information Service (NTIS) where it will be available to the general public, including foreign nations.**

**This report has been reviewed and is approved for publication.**

**JEREMY HENDRIX, Captain, USAF**
**Project Scientist**

**DEE H. ANDREWS**
**Technical Advisor**

**CURTIS J. PAPKE, Colonel, USAF**
**Chief, Warfighter Training Research Division**

**Contract Number: F33615-01-M-6049**
**Contractor: Protobox LLC**

# REPORT DOCUMENTATION PAGE

*Form Approved OMB No. 0704-0188*



**1. REPORT DATE** *(DD-MM-YYYY)*
September 2002

**2. REPORT TYPE**
Final

**3. DATES COVERED** *(From - To)*
March 2001 to February 2002

**4. TITLE AND SUBTITLE**
An Embedded Simulator Test Evaluation Monitor (ESTEEM) to Improve Distributed Mission Training

**5a. CONTRACT NUMBER**
F33615-01-M-6049

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
65502F

**6. AUTHOR(S)**
Ronald B. Ewart
Gary J. Slutz

**5d. PROJECT NUMBER**
3005

**5e. TASK NUMBER**
HA

**5f. WORK UNIT NUMBER**
1M

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Protobox LLC
1464 North Broad Street
Fairborn, Ohio 45324

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Training Research Division
6030 South Kent Street
Mesa AZ 85212-6061

**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFRL

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
AFRL-HE-AZ-TR-2002-0066

**12. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release, distribution unlimited.

**13. SUPPLEMENTARY NOTES**
This research was conducted under the Small Business Innovation Research (SBIR) program as a Phase I effort.
Air Force Research Laboratory Technical Monitor: Captain Jeremy Hendrix, AFRL/HEA, (480) 988-6561, ext 227, DSN 474-6227.

**14. ABSTRACT**
Report developed under SBIR contract for Topic AF01-061. The Embedded Simulator Test Evaluation Monitor (ESTEEM) is an innovative network simulation performance monitoring system that will enable researchers to understand and quantify the performance of simulation while it is being conducted. ESTEEM will measure simulation latencies and accuracies, identify and pinpoint sources of problems, provide status and entity information, and immediately display information to the researcher. ESTEEM is based on a multiprocessor computer running the Linux Operating System with a real-time extension incorporated to provide deterministic performance. A global positioning system (GPS) capability provides accurate time-stamping of each piece of data and correlation of data gathered at multiple simulation nodes. A variety of data-gathering subsystems enables ESTEEM to measure simulator signals. The system will enable researchers to conduct experiments evaluating the interactive performance of network simulations, human pilots, and simulation participants.

**15. SUBJECT TERMS**
Distributed mission training; DMT; Embedded Simulator Test Evaluation Monitor; ESTEEM; Flight simulation; FY01 SBIR; Network simulation; Performance measurement; SBIR Phase I; Simulation performance

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UNLIMITED | 42 | Ms Liz Casey |
| | | | | | **19b. TELEPHONE NUMBER** *(include area code)* 480.988.6561 x-188 DSN 474-6188 |

**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**

# TABLE OF CONTENTS

# AN EMBEDDED SIMULATOR TEST EVALUATION MONITOR (ESTEEM) TO IMPROVE DISTRIBUTED MISSION TRAINING

## ESTEEM Program

### Scope

This report describes Phase I of the "Embedded Simulator Test Evaluation Monitor" (ESTEEM) Small Business Innovative Research (SBIR) project. The objective of Phase I was to develop the concept of a system that can provide performance measurements on a networked system of flight simulators and provide latency and accuracy data back to a researcher located at one of the sites. The contract was awarded to Protobox LLC as a nine-month effort to define the ESTEEM system and demonstrate key concepts of the system. A demonstration prototype was developed using a dual-processor Linux machine. This prototype included the basic ESTEEM concepts including a Linux real-time extension that makes deterministic timestamping possible. Test measurements of this deterministic performance are documented in this report.

## ESTEEM Overview

ESTEEM is an innovative network simulation performance monitoring system that will enable researchers to understand and quantify the performance of the simulation while it is being conducted. ESTEEM will measure simulation latencies and accuracies, identify and pinpoint sources of problems, provide status and entity information, and immediately display information to the researcher. There are two versions of ESTEEM: the standard version and a low-cost Mini-ESTEEM version called "Min-E." The standard ESTEEM is based upon a multiprocessor computer running the Linux Operating System with a real-time extension incorporated to provide deterministic performance. A global positioning system (GPS) capability provides for accurate time-stamping of each piece of data and correlation of data gathered at multiple simulation nodes. A variety of data gathering subsystems enable ESTEEM to measure simulator signals. ESTEEM measures analog and digital simulator signals, the horizon attitude directly from video presented to pilots, aircraft state data via a reflective memory interface, and network traffic including High Level Architecture (HLA), Distributed Interactive Simulation (DIS), and Distributed Mission Training (DMT) Portal capability. The Min-E is based on a notebook computer with a subset of the standard ESTEEM interfaces. It too runs the Linux operating system with real-time extension and its software is nearly the same as the standard ESTEEM's. The Min-E is intended as a low-cost implementation and used where fewer simulator signals need to be analyzed. Both systems communicate between each other across the simulator network. Two innovative techniques allow data to be gathered and returned to the researcher without impacting the simulation or network performance. ESTEEM will enable researchers to conduct experiments evaluating the interactive performance of network simulations, human pilots, and simulation participants. ESTEEM will be a powerful tool supporting the improvement of network simulation and DMT.

Figure 1 shows the ESTEEM operating concept. The researcher located at one simulation node sets up the test and specifies the data to be collected by ESTEEM at local and remote simulator nodes. Once the test has been set up, data is passively sent by ESTEEMs located at various nodes throughout the DMT system. Standard ESTEEM systems are located at critical nodes. Min-Es are located at less critical network nodes that require a subset of the data to be collected. All ESTEEMs collect data continuously and transmit that data back to the researcher's display. The ESTEEMs transmit the data during lulls in network traffic. A correlated set of data is displayed to the researcher in near-real time. During critical experiments, the researcher identifies the period of the experiment and sets up all ESTEEMs across the network to begin collecting data. The ESTEEMs collect the data passively at each site and wait until the experiment period is over before transmitting the data back to the researcher. In this mode, the researcher can be assured that the data collection did not affect the performance of the simulation system.

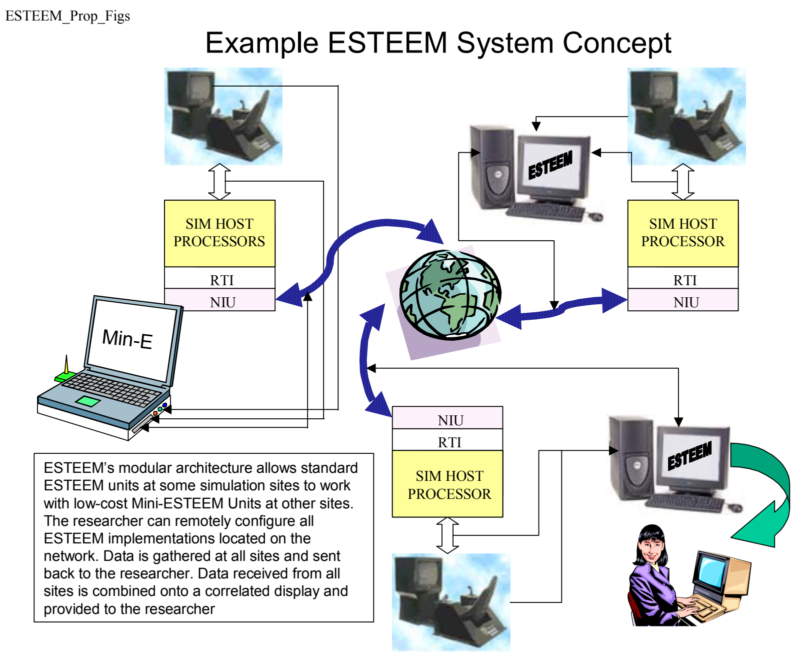



Figure 1. Concept diagram for the ESTEEM system.

**ESTEEM technical paper**

A technical paper on ESTEEM was jointly authored by Protobox LLC and the Air Force Research Laboratory, Warfighter Training Research Division (AFRL/HEA) during Phase I and presented at the Spring 2002 Simulation Interoperability Workshop (SIW) conference. The paper provides a good overview of the ESTEEM project and will be published separately.

**Significance of the problem**

The ESTEEM system is being developed so that researchers can understand the performance of their network simulations while they are being conducted. Rapid development of the DMT network has outpaced researchers' capabilities to measure and verify its performance. With several high-fidelity simulator systems slated for installation on the DMT network within the next year, it is extremely important that researchers have the tools to measure the performance of the network and the end-to-end performance of the simulations conducted on that network.

The significance of the problem has not diminished. If anything, the potential problem has gotten larger. At the time the Phase I proposal was written, the Air Force as well as the entire Department of Defense (DoD), had mandated the exclusive use of HLA and its Defense Modeling and Simulation Office (DMSO)-developed Run-Time Infrastructure (RTI) as the only way in which to network DoD simulations. Since that time, practical experience has exposed the large amount of computational overhead associated with the use of the RTI. Several organizations, including AFRL/HEA, have recognized the practical difficulty of interfacing legacy simulations that are still fully functional, but which have been developed using the prior DIS standard, or other interface techniques. Some organizations have no intention of scrapping the investment they have made in their existing simulations. They seek to develop interfaces to transform their network data into HLA's RTI protocol via bridges or other translational units. Any type of bridge or transformation solution requires verification to assure valid performance

The problem has also grown for the Air Force's DMT network itself. TRW, the Operations & Integration (O&I) contractor responsible for developing the network infrastructure has developed the "Portal" concept. The portal is essentially a computational unit that sits at each simulation node and "translates" different protocols into a common network protocol. It has the positive aspect of allowing simulations that operate using different protocols to be compatible on the common DMT network. The negative aspect is that the Portal introduces yet another source of latency at each side of the simulation network. Latency is the enemy of a quality simulation. The Portal formats outgoing messages from the local site into the common DMT protocol. It also converts the common DMT protocol that represents all of the entities of interest into the protocol used by the local simulations. There is a potential for latency problems to be introduced by this technique. For example, the conversion efficiency may be different for each different protocol converted. And there is always the potential for problems as the network traffic is increased. The DMT concept is to use the network for ancillary traffic in addition to the typical entity traffic. For example, communications, which consumes a significant amount of bandwidth (even using compressed modes), can rapidly eat up precious bandwidth. The researchers need ESTEEM to measure the interactive performance of simulations participating on the DMT network, and to assure that results obtained on their DMT network are valid.

DMSO's approach to HLA's RTI development also heightens the need for ESTEEM. The RTI is very robust and can handle a variety of data transmission requirements. It is also very object oriented. Library calls are used throughout the architecture. While this approach provides a means of accommodating a wide range of simulations from "constructive" to "virtual," it does so at the expense of protocol efficiency. The RTI takes significant computational processing power and can result in unexpected latencies being introduced between the simulation and the network. The RTI problem has been complicated by the fact that different builds (i.e., different implementations compiled for a specific operating system) of the RTI have been developed for various operating systems. The Viper simulation at AFRL/HEA uses the VxWorks operating system.

The RTI implementation at different simulation nodes can often be running on different platforms under different operating systems. The result is that each node may have different performance. Each RTI implementation can introduce different latencies. The researcher can no longer be satisfied with understanding the performance of the local simulation. When the researcher is conducting a multinode experiment across a DMT network, he must understand the interactive performance of the entities he is working with while the simulation is in progress.

Researcher's can use ESTEEM to quantify the performance of any company's RTI. ESTEEM will allow them to measure simulation latency and accuracy and get a good understanding of the strength and weakness of each RTI.

ESTEEM will also be important from a "day-to-day" operational standpoint. Experiments that do a rigorous verification of network and simulation performance are done infrequently. In the past, these tests were done with equipment such as the original Simulation Network Analysis Project (SNAP) equipment developed by AFRL Air Vehicles Directorate (AFRL/VACD). The SNAP equipment was installed for a particular test, data were collected, and then a lengthy period of data reduction followed. SNAP passively gathered data at multiple sites, and then the data were combined at a single site using manpower-intensive techniques. Often by the time the data were reduced, a week or two passed and the researchers who could have used the data were working on new projects.

ESTEEM has two important features that correct problems associated with the original SNAP equipment. First, it is an embedded tool. Once installed on a simulator, ESTEEM can remain connected to the system at multiple nodes, and it can be used anytime the researcher needs the information. The researcher does not need to plan for a special installation weeks ahead of an experiment.

Another ESTEEM feature of importance is that it can be used to continually monitor the health of the DMT network on a day-to-day basis. Once a DMT configuration has been baselined, maintenance personnel can use ESTEEM to perform a daily readiness check. The performance of the DMT network is not fixed, but is dynamic and unpredictable. With additional simulators being added over time, and with potential changes to the backbone or interface at various sites, ESTEEM can tell simulation personnel when the DMT performance has changed, or when it has degraded to a point that it is likely to affect training.

Full development of ESTEEM in Phase II will, for the first time, enable researchers to understand the quality of their interactive simulations as they are being executed. The system will provide the researcher with information in researcher-configurable plots and related reports.

The proposed ESTEEM simulation tool will enable researchers to improve the quality of training and research that they conduct via network simulation. If the quality of the tested simulation is high, researchers will have confidence in the results of their human experiments conducted on those simulations.

## Requirements definition

The Phase I SBIR project began with a review of the proposed approach and definition of the system requirements. Protobox is very concerned about the deterministic performance of the ESTEEM system. The system must be able to handle many interrupts simultaneously without varying the accuracy of the time stamp. As a result of the system requirements definition, the following requirements have been identified. The requirements and the goals for Phase I and Phase II of the SBIR program are identified in Figure 2.

| Requirement | Phase I | Phase II |
|---|---|---|
| Deterministic performance (interrupts) | Accurate to 0.1 ms | w/GPS time tag |
| HLA compatibility | Selected variables | All FOM variables |
| Flexible Input / Output architecture | DI, DO, AI, AO | DI, DO, AI, AO |
| Ethernet data collection | 1 | 2 |
| End-to-end data collection | Stick-to-visual single ship | Stick-to-visual multi-ship |
| Data acquisition | Key demonstration variables | Any variable |
| Expandable architecture | Yes | Yes |
| Human factors GUI; clear concise display to researcher | Defined; partially implemented | Implemented |
| Human factors experiment | Defined | Implemented |
| No impact to network during critical measurements | Demonstrated | Implemented |
| Minimum impact to network during data acquisition | Demonstrated | Implemented |
| Remote data collection | Best approach studied | Implemented |
| Compatibility with common analysis tools | Demonstrate capability | Implemented |

Figure 2. ESTEEM Requirements

## Electronic Visual Display Attitude Sensor development

One of ESTEEM's strengths is its ability to measure a variety of cues presented to the simulator pilot. Synchronization of cues presented to a pilot has historically been a problem. Time delays, which are introduced by computer-generated image systems, are always a concern for high-fidelity simulations. The out-the-window visual display that is generated by the computer image generator must be correlated with the simulation and input/output routines in order to properly synchronize the cues.

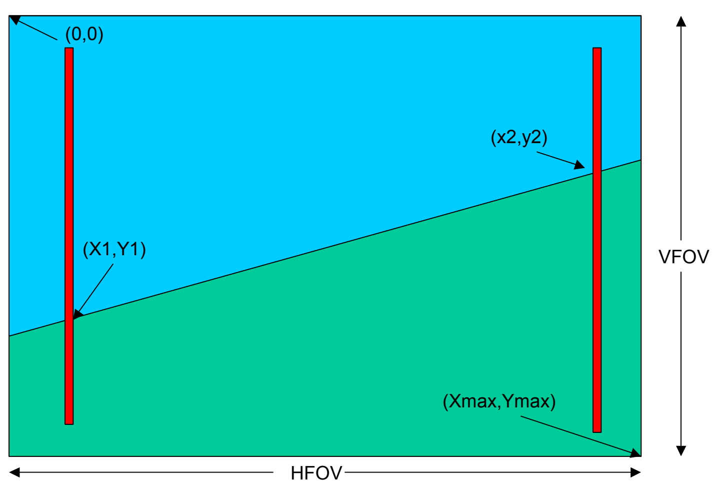
ESTEEM incorporates an Electronic Visual Display Attitude Sensor (EVDAS) to measure pitch, roll, and attitude angles of features in a video signal in raster format being presented to the pilot.

EVDAS measures and outputs horizon angles deterministically at the end of each video field. It has additional modes that allow it to measure angles of graphical features such as a rung on the pitch ladder of a heads-up display (HUD).

Protobox designed and prototyped an improved EVDAS system that incorporates the best features developed previously and includes some additional features to make setup easier. The new features include an automated color setup feature, and a faster automatic line-rate synchronization scheme. The new EVDAS will also run at a significantly higher clock speed than previous implementations, so that the horizon angles generated by EVDAS will be more accurate. The new design will be implemented on a single card that can be interfaced to either a desktop or notebook computer.BNC connectors interface EVDAS to the red-gree-blue (RGB) signals going to the pilot's display. Like all preceding EVDAS implementations, EVDAS generates an interrupt on the first line of the vertical interval. The horizon angle data is available at that time for transfer from EVDAS to the host computer. The interrupt also corresponds with the accepted definition for image generation (IG) latency widely accepted in the visual system community; i.e., the end of the active video field.

EVDAS is typically connected to the video going into the pilot's visual display; it also interfaces to ESTEEM for input/output (I/O) and setup data transfer. For purposes of simplification, this report will illustrate sensing of a blue sky and horizon although EVDAS can be used to sense other visual images. For example, it can measure a HUD's pitch ladder position or a lead aircraft's roll angles as well. EVDAS electronically generates two movable vertical sensing stripes or windows in video. One stripe is placed near the left side of the displayed image and one is placed near the right. These electronic stripes are simply gating pulses that tell EVDAS when to sense for the presence or absence of sky. Figure 3 illustrates the typical EVDAS setup as it measures a horizon roll angle.

EVDAS electronically measures the X-Y pixel coordinate of the intersection of the horizon with each of the two vertical sensor lines. In Figure 3, the left point is (X1, Y1) and the right point is (X2, Y2). Assuming that the display is linear, i.e., non-distorted, the horizon roll angle can be calculated as illustrated in Figure 3. Likewise the pitch angle is similarly calculated. In practice, EVDAS computes both pitch and roll angles each video field time. This data is calculated during the active field time and output to ESTEEM on the first line of the vertical interval.




Simplified Roll Angle Calculation

$$\text{Ø} = \text{Arctan} \frac{(Y2-Y1)*(VFOV/Ymax)}{(X2-X1)*HFOV/Xmax}$$

Figure 3. Typical EVDAS visual display roll angle calculation

Figure 4 is the block diagram of the new EVDAS system developed during Phase I of the ESTEEM program. It uses one of the newest Analog Devices (A/D) video converter with a built-in Phase Locked Loop (PLL) to digitize the video and synchronize EVDAS with the video raster.

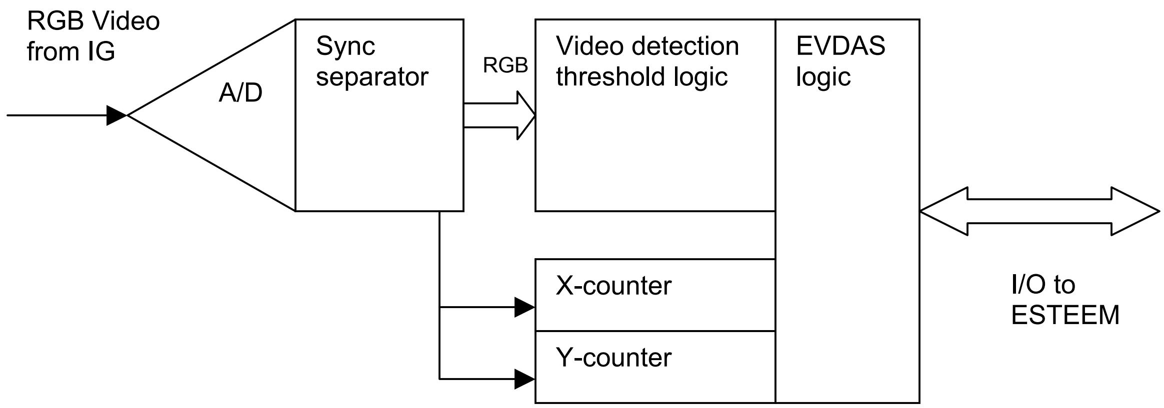



Figure 4. EVDAS block diagram

For the first time in this EVDAS implementation, the sensing stripes will be extremely narrow. The sensing stripes will now be a sensing line that is only one pixel wide. By reducing the width of the sensing stripe, accuracy of the data will be significantly improved. There will be no ambiguity of whether the data came from the left or right edge of the sensing stripe. The new narrow sensing stripe will give a much more precise data point, and therefore result in a more accurate sensed horizon angle

EVDAS was designed and prototyped during Phase I. The EVDAS implementation included most all of the EVDAS logic implemented in an Altera Apex EP20K100QC208-1X FPGA (Field Programmable Gate Array). Protobox LLC used Altera's "Quartus II" development environment to develop the configuration program for this chip. The remainder of EVDAS including the analog video processing section and the passive components was designed using Orcad Capture. Orcad Layout was used to design the Printed Circuit Board (PCB).

**Mini-ESTEEM**

The biggest change in approach that occurred as a result of Protobox's Phase I research was to abandon the "agent" approach for low-cost ESTEEM implementation in favor of development of a Mini-ESTEEM system. This decision was made for several technical reasons. The original "agent" approach was problematic from several aspects. The amount of useful information was limited using the "agent" approach, and the approach was highly dependent upon the host simulator's computer architecture. The "agent" approach also required that each simulator to be measured include the Java virtual machine installed on it. This JAVA virtual machine consumes processing power and could impact simulation performance depending upon its implementation; this violated one of the primary ESTEEM principles, which is not to impact the simulation performance. Time stamping of critical data was also difficult because a common GPS reference could not be accessed for many of the simulators of interest. Use of simulator-unique time references would have invalidated the otherwise deterministic ESTEEM data. Interfacing the signals was unique for each simulator to be analyzed. This would require ESTEEM software to be modified for each simulator implementation.

The new Min-E has been substituted for the original "agent" approach. Both approaches achieve a low-cost implementation of ESTEEM; however, the Min-E has many advantages over the "agent"-based implementation. The Min-E is simply a single-processor notebook computer version of the larger multiprocessor standard ESTEEM system. The Min-E uses 90% of the ESTEEM software so very little additional coding is necessary. It includes a subset of the full

ESTEEM capabilities including a GPS clock, analog and digital I/O, EVDAS, and a single Ethernet card. The Min-E is very portable and unlike the agent implementation includes its own integrated display screen (see Figure 5).

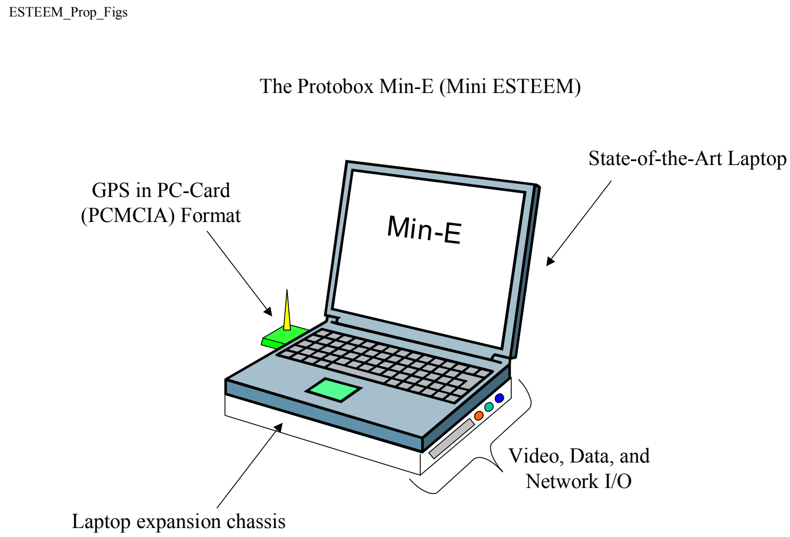



Figure 5. Mini-ESTEEM

As in the main ESTEEM system, the Min-E runs on the Linux Operating System and uses the exact same deterministic technique to time stamp the data. The primary limitation of the Min-E system is that it is a single processor implementation, has only a single Ethernet interface and has a fewer number of interface channels. Otherwise the architecture is identical to the ESTEEM system, and very little additional software development will be required to implement the system.

The goal of the Min-E development is to design a low-cost implementation of the ESTEEM system, which has nearly identical software and a significant subset of ESTEEM's full-featured implementation. During Phase II, the Mini-ESTEEM notebook computer will be selected along with the individual interface components. Wherever possible, the interface hardware will be similar or identical. For example, a different version of the National I/O card will be used for the Min-E; however, the hardware calls will remain unchanged, and only the I/O card driver will be changed. Similarly, the EVDAS card will be the identical card, and no additional software is anticipated to be required.

**TRW Portal Investigation**

TRW is the Operations and Integration (O & I) contractor responsible for development and support of the DMT network. As such, they are developing concepts so that the various simulation systems can operate with each other.

TRW's approach is to develop a "Portal" to allow various protocols and devices to coexist and operate efficiently on the DMT network. The DMT network uses a DMT common format running on an ATM service and Portals located at each simulation site. The Portals translate the DMT common format to and from the protocol used by the local simulation network. Typically these local simulations will be running either DIS or an implementation of HLA such as DMSO's RTI 1.3v6 or 1.3 NG. Initial investigations are ongoing at Protobox to determine the best way for the ESTEEM system to communicate across this DMT network. The DMT simulation network will be an important simulation tool in the next few years. If ESTEEM is to be successful, it must be capable of working on local area networks (LANs) operating with a single protocol, as well as the DMT simulation network operating with multiple protocols using the Portal concept.

Initial contact is being made with representatives of TRW to coordinate the development efforts of ESTEEM to ensure that it is compatible with the Portal concept being developed by TRW for the DMT network. Some of the information regarding the Portal concept was determined from the "Standard Portal Interface Specification," and additional information was learned from position papers written by TRW. Figure 6 shows the top-level Portal concept.

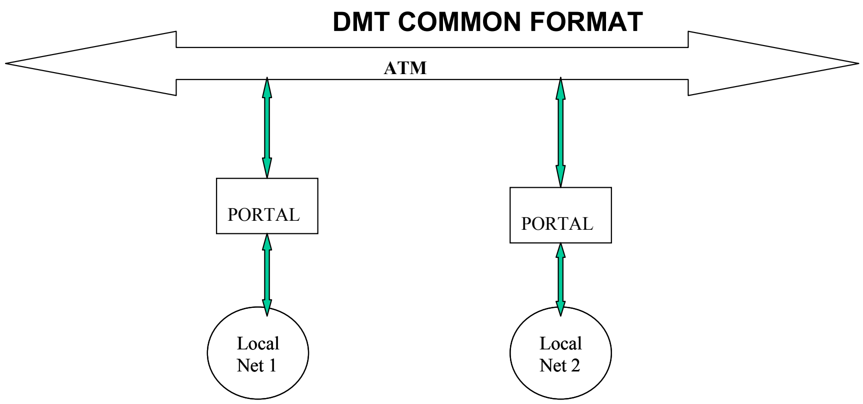



Figure 6. TRW Portal concept

Figure 7 shows how the Portal will interface the typical DMT network to the local Federate Simulation System LAN. The KG-175 TACLANE encryption device is used to encrypt the data prior to transmission on the DMT network. The O&I contractor will configure the Portal to properly interface with the "Red" side of the TACLANE for the Federate Systems. All of the units and processes illustrated in Figure 7 have some latency associated with them. ESTEEM's capability of measuring simulation latency and accuracies between multiple sites on a long-haul network will be beneficial to optimize the DMT system performance.

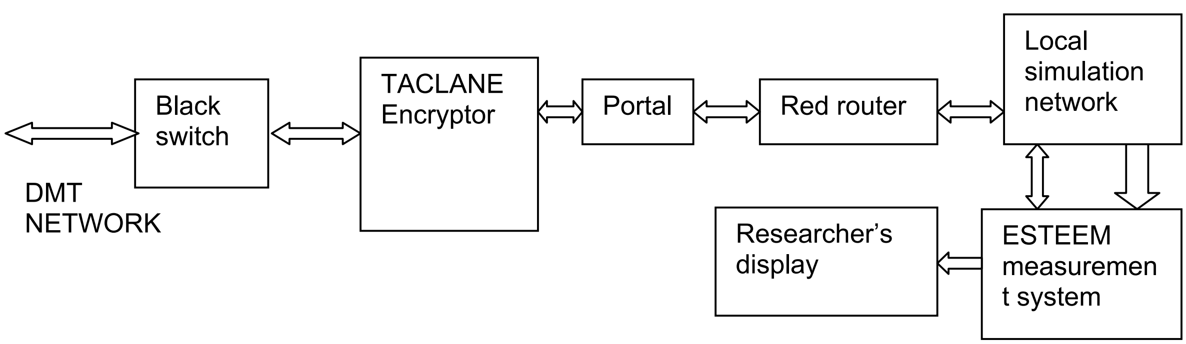



Figure 7. Portal concept

The Portal is the interface between the local simulation network and the external DMT network. As such, it translates various local simulation protocols into a "DMT common format" protocol for transmission to the remainder of sites on the O&I network. The Portal looks like the combination of the entities at the local simulation site to the external network. Likewise, the Portal looks like all of the remaining entities to the local simulation network. The local Portal translates simulation data that leaves one simulation site into the "DMT common format." It is then re-translated into the same or perhaps different format at each of the other simulation sites. One potential difficulty with this approach is that a bit of information that leaves one site does not necessarily end up as the same bit of information at the second site.

The Portal Specification indicates that the Portal will be compatible with several different protocols. ESTEEM needs to be able to send simulation performance data between several ESTEEM units located at various nodes in the simulation. The concept is to passively collect simulation data at multiple sites, and then send the data to the researcher's site located at one or more network nodes. The researcher needs to be able to understand the performance of the network while the simulation is ongoing; however, the data does not need to be sent at high priority between ESTEEM nodes. The goal is to present the interactive simulation performance to the researcher within a few seconds of its occurrence. The ESTEEM system will be able to precisely gather the data and correlate each piece of data precisely with GPS time stamps. Once the piece of data has been accurately time stamped, the amount of time it takes to transmit the data to the researcher's site is less critical. The goal is to send the data back during nonpeak network usage moments. The other ESTEEM concept is that no data is put on the network during the few seconds of critical experiments. This is done to prevent suspicion that the introduction of ESTEEM data on the network is in some way altering the performance of the simulation itself. By eliminating ESTEEM data transmission during critical experiment periods (typically a few seconds at a time), only the simulation is actively using the network.

The Portal specification requires it to be compatible with several different protocols and data formats. Figure 8 shows some potential ways that the ESTEEM data could be transmitted from one ESTEEM unit to another.

| **Protocol or data format** | **Comment** |
|---|---|
| DMT DIS standard PDUs | A DIS packet could be developed to transmit ESTEEM data (see discussion below). |
| DMSO HLA RTI 1.3v6 and 1.3NG | Not a good candidate because it requires other entities on the net to be aware of traffic |
| FTP | |
| TFTP | |
| TELNET | |
| HTTP | Potential, but requires more bits to represent ESTEEM data |
| SNMP (Simple Network Management Protocol) | |
| H.323 Video Teleconference Packet | Not applicable |

Figure 8. Potential protocol or data format for ESTEEM inter-system communication

There are several issues that need to be addressed before selecting a data format for transmission between the ESTEEM units. Figure 9 includes these considerations:

- The data transmission technique needs to be compatible with a DIS network, an HLA RTI 1.3 network, and the DMT Portal concept.
- Transmission of ESTEEM's data must not affect other simulations or entities on the network.
- The goal is to select a protocol or data format with a minimum overhead associated with the data transmission. This is required so that the ESTEEM data does not affect other simulations.
- The Portal must be able to be configured so that the data packet can be transmitted from one site to another and get through to the local simulation network.

Figure 9. Data protocol / format requirements for ESTEEM data transmission

At this stage of the investigation, a DIS packet tailored for an ESTEEM data transmission appears to be the most likely choice. One of the Portal requirements for the Portal as well as the networks attached to the Portal is that they be able to operate properly with "receipt of non-applicable PDUs and the non-applicable PDUs must not interfere with normal operation." Additional work is needed to define the best way to transmit data between ESTEEM units.

**ESTEEM Program Status**

The Phase I ESTEEM effort concluded on February 17, 2002. Although a Phase II proposal was presented to AFRL/HEA, this proposal was not funded. Protobox LLC hopes that work on the ESTEEM system can continue in some form, and that ESTEEM will someday become a key element in the development of the Air Force's DMT simulation network.

## Software Development

**Operating System Selection**

The first step in the selection process was to establish the basic requirements for the ESTEEM operating system (OS). These requirements are summarized in Figure 10.

| | |
|---|---|
| Must support hard real-time performance. | Required |
| Must support HLA/RTI | Required |
| Must provide a user-friendly GUI environment. | Required |
| Must support multiple processors. | Required |
| Must support multi-I/O cards | Required |
| Must support multiple Ethernet cards. | Required |
| Must support GPS card. | Required |
| Must support Scramnet interface or equivalent | Desirable |
| Must have source available | Desirable |
| Must run on an x86 platform | Desirable |

Figure 10. Operating system requirements

The foremost requirement is to support hard real-time performance. It is key that a system designed to provide timing information not suffer from any variabilities in its architecture. Hard real-time means guaranteed response time. This guaranteed response time is a function of design, not chance. Some pseudo real-time systems make claims of typical response times. These systems suffer from the occasional "milliseconds" glitch. This is not acceptable for a precise measurement system.

The OS must also be capable of providing a Graphical User's Interface (GUI) environment for monitoring and control of measurement processes. These GUI's typically run in a nonreal-time mode. The OS must therefore support hard real-time and a user-friendly GUI environment simultaneously. The use of two separate operating systems to accomplish this goal was not considered acceptable.

A High-Level Architecture Real-Time Infrastructure (HLA/RTI) will run on ESTEEM. The OS must therefore be on the list of DMSO RTI-NG 1.3 v3.2 supported operating systems. As of June 20, 2001, that list included:

1. Linux 2.2 Redhat 6.1 x86 architecture
2. Sun Solaris 2.7
3. SGI Irix 6.5.6
4. VxWorks 5.3.1
5. Windows 2000
6. Windows NT
7. Windows 98 SE.

The OS must support multiple processors. Protobox believes that it will require more than one processor to meet the real-time requirements coupled with the ability to run an RTI and data presentation software. The preliminary design will have the RTI dedicated to one processor while the real-time data acquisition and GUIs run on the other.

The hardware will contain a multichannel analog and digital I/O card, GPS, EVDAS, Ethernet cards and possibly some type of shared memory card. This OS must be capable of supporting this hardware. The OS must support multiple Ethernet cards. Some simulation architectures employ devices (such as Network Interface Units [NIU] or encryption units) that have an Ethernet input and output. ESTEEM must be able to simultaneously be able to monitor the traffic into and out of the device in order to measure the delay through the device.

It is considered highly desirable to have the source code for the operating system and its extensions. It is anticipated that some customization of the OS may be required to achieve the ESTEEM measurement goals. Protobox engineers have many years of experience with real-time operating systems and are very comfortable with the idea of making modifications to the OS.

The OS must run on x86 architecture. Though this is not mandatory, it is highly desirable. The x86 architecture is widely supported in both the software and hardware arenas. The ability to use commercial off-the-shelf (COTS) software and hardware products in the design of ESTEEM simplifies the design and reduces the costs.

The list of possible operating systems was narrowed down to either Windows NT/2000 or Linux based on the above requirements. Linux was chosen as the best compromise for ESTEEM's requirements.

### Linux

Linux supports many of the requirements as delineated in Figure 10. A Posix-compliant Linux kernel provides some of the basic features required for hard real-time performance. The Posix patches to the basic kernel provide prioritized preemptive scheduling as well as processor affinity and memory lock-down support. A problem with Linux is that the kernel itself is not totally pre-emptible. This can result in having long periods of time where interrupts are blocked.

It is necessary to employ a real-time Linux extension to achieve hard real-time performance. The Linux real-time extensions work in a very similar manner to the Windows extensions. A separate real-time kernel is used to schedule real-time processes and control interrupts. The source code for Linux extensions, like Linux itself, is readily available.

Windows is clearly more mature than Linux in the areas of off-the-shelf GUI applications and enjoys more support in terms of drivers for new hardware. The key factor for ESTEEM however is real-time performance. Both operating systems require real-time extensions to achieve hard real-time performance. As previously stated, it is anticipated that some customization of the OS may be required to achieve the ESTEEM measurement goals. This is only possible if the source code is available. For this reason, Protobox has decided to use the Linux OS with a real-time extension. Linux real-time extensions are discussed in the paragraph below.

### RT Extension Selection

The two primary candidates for the Linux real-time extension were RTAI and RTLinux. Both extensions use the same approach. RTAI is actually an outgrowth of RTLinux. The RTLinux philosophy is to keep the real-time kernel "lean and mean" whereas RTAI's approach is to add more features (and possibly more bugs) to the real-time kernel. RTLinux seems to be more popular at this time and has a wider support base. For these reasons Protobox has chosen RTLinux for the real-time extension.

RTLinux is based on a "virtual machine" approach. It serves as the virtual machine for Linux. All interrupt control instructions are intercepted by RTLinux preventing Linux from blocking interrupts and holding off real-time processes. The RTLinux scheduler is an installable Linux module. The scheduler runs Linux as the idle process.   All RTLinux processes run in kernel space and in fact are also installable Linux modules.

An RTLinux process can be scheduled via several different scheduling schemes or can be connected to run off an interrupt.   RTLinux provides FIFO's and a shared memory module to implement communications between RTLinux tasks and Linux tasks.  Figure 11 shows the relationship between the different software pieces when using RTLinux for a real-time data acquisition task.

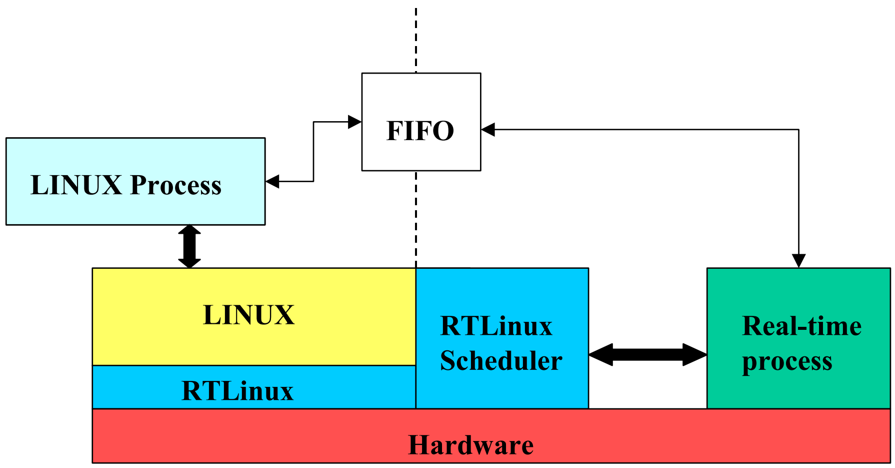



Figure 11.    RTLinux software architecture

**Computer Selection**

The Phase I ESTEEM computer was selected based primarily on its compatibility with the Linux operating system, its ability to handle interrupts, and its support of multiprocessors. The computer architecture needed to be common enough that code developed on it during Phase I could be subsequently moved to a higher power computer. A Dell Precision 530 Workstation was selected. This computer system has two 1.4 GHz Xeon processors and a front side bus that operates at 400 MHz.  The computer uses 512 MB of high performance RIMM Rambus Memory.  The computer runs the Linux 7.1 Red Had operating system.  The real-time extension to Linux is discussed separately in this report.

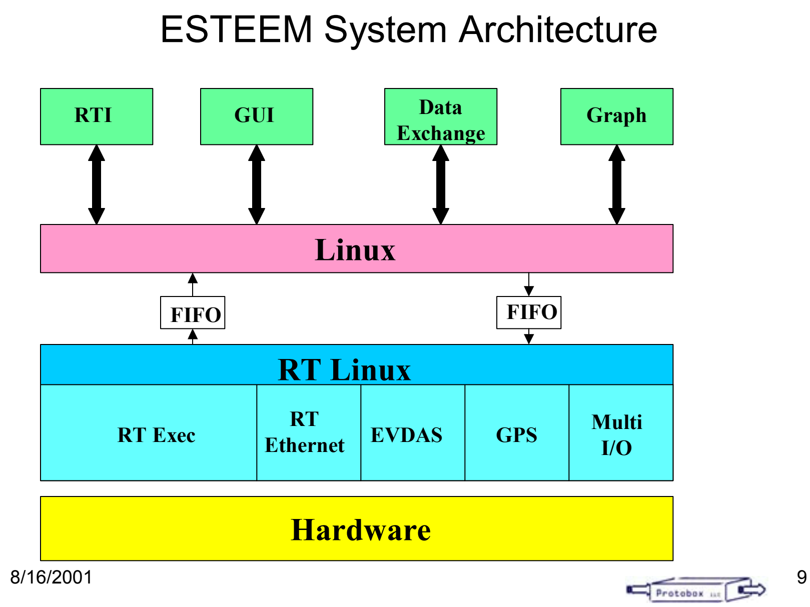



Figure 12. ESTEEM system architecture

**Multifunction Input / Output Interface Card**

A data collection technique is to use analog and/or digital signals from the simulator under test. For example, stick inputs representing force or position may be measured. The National Instruments Model NI 6036 was selected for this purpose. The NI 6036 has 16 single-ended analog inputs. The card samples the analog inputs at 200 KS/s at a 16-bit resolution. The card also outputs two analog outputs also with a 16-bit resolution. These outputs will be used to drive various simulator points during special tests. For example, pilot inputs to stick or other inputs will be repeatable output via these channels. The NI 6036 also has 8 digital I/O lines which can be used to sample various switch positions or other digital signals. Two 24-bit, 20 MHz timers are also available on the card. The card uses external digital triggering.

**Interrupt handling concept**

Interrupts are received by the ESTEEM system from several sources during recording of simulation data. These interrupt sources include the host-computer synchronizing pulse, Ethernet interrupts, hardware I/O interrupts, timers, and the EVDAS that interrupts ESTEEM on the last video line of each video field. Any combination of these input interrupts could fire their interrupt simultaneously. The problem is to define a technique that will handle multiple interrupts deterministically.

When any interrupt occurs, the ESTEEM computers need to time stamp the data and process the data in some manner. Some of the data processing is relatively easy. For example if an interrupt occurs that needs to read a digital input register, all that needs to happen is a very short software procedure to read a particular external port. On the other hand, some of the processing requires a good bit of software processing. For example, when an Ethernet interrupt occurs, there is a good bit of handshaking that needs to occur to read a FIFO buffer that may be hundreds of words long. Some filtering of the data may also be required.

If four interrupts were handled sequentially, the time stamp for Interrupt 4 could be measured a significant amount of time after the time stamp that is attached to the first interrupt, even though it occurs at the same time.

The goal is to define an interrupt handling technique that will deterministically time stamp and process multiple interrupts if they occur simultaneously without compromising the integrity or accuracy of the data or time stamp.

The following technique has been developed to accurately time stamp and process simultaneous interrupts. For each interrupt process, the process is broken down into two components. The first is a very short process that simply time stamps when the interrupt occurs but does not do any processing of the data. The second component actually processes the data. This second process may vary in length. If multiple interrupts occur simultaneously, or very closely to each other, only the first “time stamp” portion of the processing will be performed. When all of the interrupts have been time stamped, the longer processing component will be handled. This technique ensures that all data is deterministically time stamped.

Typically data in flight simulators is valid and can be processed slightly behind when the interrupt occurs. For example, it is important to time stamp an Ethernet packet when it arrives, however, the FIFO buffer retains the data until it is read out by the computer.

**Software Development**

Key elements of the ESTEEM software were developed during Phase I. These elements were selected based on their criticality to the ESTEEM system and their risk factor. For example, ESTEEM’s deterministic performance with multiple interrupts is considered critical and of high risk. Software development concentrated on this area. Other areas such as GUI development were considered low risk and deferred to Phase II development. Some of the key software development areas are discussed below.

**Development of the 2.4.4 kernel**

One of the unique features of Linux is the ability to customize kernels by building them from source code. Protobox engineers have been gaining experience in this process by building several versions of the 2.4.2 kernel shipped with the Dell multiprocessor system. RTLinux version 3.1 real-time extension requires the 2.4.4 Linux kernel. The source code for this version of the kernel was located and downloaded. The first step in the process of building a new kernel was to generate a configuration file and customize it to Dell computer platform; this is the platform that the kernel will be running on. The configuration file is also used to establish which drivers/modules will be an integral part of the kernel and which will be installable modules. The next step was to build the kernel and any installable modules. The new kernel was then booted and special installable modules were built. The driver for the NVIDIA graphics card falls into this special category for ESTEEM and must be built after the new kernel is booted. The 2.4.4 kernel was successfully built and tested using the preceding method.

**Integration of RTLinux real-time extension and development of kernel**

The RTLinux real-time extension consists of patches to the kernel and several installable modules. The first step in installing RTLinux was to apply the patches via the “patch” utility. This utility makes source-level changes in the kernel source tree. The primary function of the kernel patches is to give control of the hardware interrupt controller to RtLinux. The modified kernel was then built using the procedure outlined above. Finally the RTLinux modules were built and run as installable modules. Protobox engineers have successfully built a real-time kernel with RTLinux 3.1 and Linux 2.4.4.

Figure 13 shows the software architecture of a RTLinux/Linux real-time system. As can be seen in this figure, RTLinux is positioned between Linux and the hardware. Any attempts by Linux to block interrupts is intercepted by RTLinux and handled in an appropriate manner. RTLinux emulates “blocking” in software so that as far as Linux is concerned interrupts are indeed blocked. At the same time RtLinux will allow real-time interrupts through to the real-time tasks. Going the other direction, RTLinux decides whether a hardware-generated interrupt should be passed on to Linux. This decision is based on the current blocking state requested by Linux and any current real-time activities.

Another key concept depicted in Figure 13 is that Linux and all of its associated processes are run as the idle or lowest priority task by the RTLinux scheduler. This permits the real-time tasks to run at the highest kernel privilege level. These real-time tasks have direct access to the hardware. By using this architecture, ESTEEM will be able to achieve its goal of deterministically time stamping each piece of data. This architecture combined with the high and low priority interrupt handling concept will enable ESTEEM to process multiple interrupts for the simulator systems without time-tag jitter.

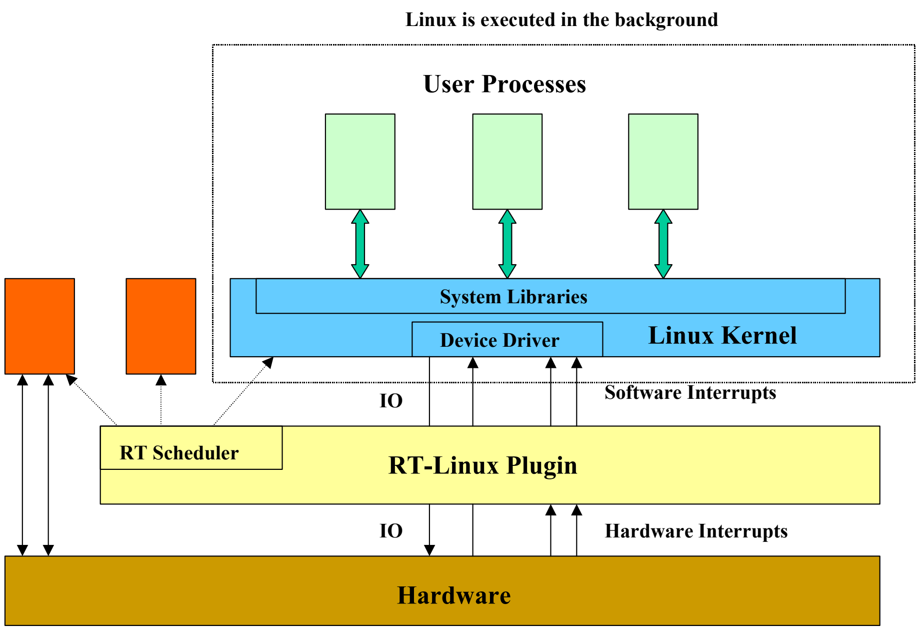



Figure 13. Software architecture of ESTEEM's RTLinux/Linux real-time system

### Multi-IO driver

Protobox installed a National Instruments PCI- 6036E multifunction data acquisition board into the ESTEEM computer. This board runs on the PCI bus and has a 200 ks/s sampling rate. It contains sixteen 16-bit A/D channels, two 16-bit D/A channels, eight digital I/O lines, and two 24-bit counters. The vendor of the board does not provide Linux drivers.

No Linux driver for the PCI-6036E was located; however a driver for the PCI- 6035E card was found at the Linux Control Measurement Device Interface (COMEDI) project site. The source code for this driver, along with a user-level library (COMEDILIb) and a kernel-level library (KCOMEDILIB), was downloaded and built. The kernel-level COMEDI software consists of a common data acquisition module (comedi.o), a data acquisition card specific module (ni_pcimio.o), and the kernel library. This library is used when another kernel module needs to make data acquisition calls. A library is also provided for user-level data acquisition tasks. Some modifications were necessary to make this driver work with the PCI-6036E card. Protobox modified and tested the COMEDI driver with the PCI-6036E board.

### Data Acquisition testing.

Protobox developed a test program to verify proper operation of the COMEDI driver with the PCI-6036E board. This program read and displayed the value of a user-selected A/D channel in volts. A variable voltage source was connected to the test A/D channel and a voltmeter was used to verify the correct readout. The program also set a D/A channel to a voltage selected by the user.

Again a voltmeter was connected to the D/A channel to verify proper operation. Finally the program set and reset a digital output line (digital lines can be configured as input or output). The digital channel was connected to +5 volts through a pull-up resistor. A voltmeter was used to verify proper operation.

**RTLinux Real-time task development**

RTLinux processes (tasks) are implemented as Linux installable modules running at kernel-privilege level. The processes can be scheduled via the RTLinux scheduler or directly connected to an interrupt. The ESTEEM architecture will require both types of modules. An additional requirement will be for the real-time processes to share data with nonreal-time tasks. RtLinux provides two methods for this communication: FIFOs and shared memory. On the Linux side, a FIFO looks like a character device and is read in that manner.

Protobox LLC designed the ESTEEM data capture process during the second reporting period. Figure 14 shows a simplified design of this process.

The interrupt handler gets a time stamp and places it in the header of a data packet. The interrupt handler will then resume a real-time task that is in a suspend state. The real-time task will get the data (read an A/D, etc.) and place the data in the packet. It will then copy this packet to the FIFO. The act of filling the FIFO will resume the user task. The user task then gets the data from the FIFO and processes it as desired.

A prototype interrupt handler was designed and coded to test the basic architecture. This handler was connected to an interrupt from the real-time clock. When the interrupt fired, the process incremented a counter and wrote the value to a FIFO. A user process then displayed the counter value. The interrupt rate was varied and the output rate of the user process was verified to be in accordance with the selected interrupt rate.

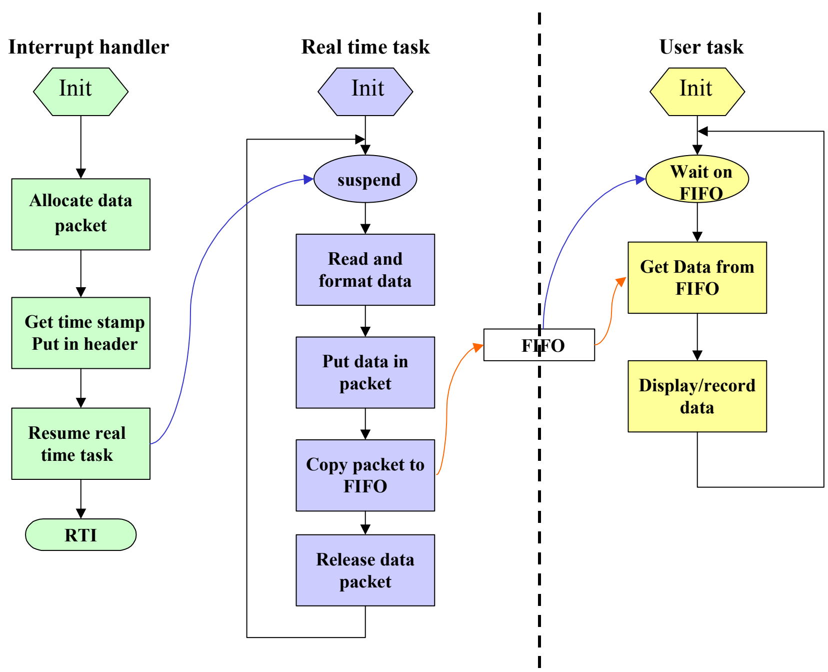



Figure 14. ESTEEM data capture process

**General Purpose Counter Software Development**

The PCI-6036E DAQ National Instruments data acquisition card contains two 24-bit counters. These counters will be used for local time stamping, interrupt generation, and timing. These counters are also the means by which external interrupts enter the system.

No user level code was available for simple reading and writing of counters. Protobox engineers developed user-level routines to configure, read, and write the counters. A user-level process was written to verify proper operation of these newly developed timer routines.

External interrupts will be required for external triggering of the ESTEEM data collection software. External interrupts enter the system via the gate inputs on the counters. The gate signal acts as a general-purpose control signal and can operate as a counter trigger signal, a counter enable, a save signal, a reload signal, an interrupt, an output control signal, a load register select signal, and a counter disarm. For external interrupts, the gate signal is configured in external mode with its input coming from a user-designated pin. The card is then setup up to generate an interrupt on the gate signal. Protobox engineers developed and tested code for the kernel library to configure and control the counters. Modules running in kernel mode that wish to interface to the DAQ card use this kernel library.

**ESTEEM Kernel Module Development.**

Protobox developed the ESTEEM kernel module, which will be used to handle real-time data collection. This module consists of an initialization section, an interrupt handler, real-time data collection threads, and a termination section.

The heart of this module is the interrupt handler. This handler is executed anytime the DAQ card generates an interrupt. The handler must first determine the source or cause of the interrupt, acknowledge the interrupt, and then take appropriate action. For the ESTEEM application the interrupt handler determines which counter caused the interrupt. It then reads that counter to obtain a time stamp. (Currently the counter is setup so that it is reset to zero when an interrupt occurs. This setup allows for direct measurement of latencies by reading the counters.) It then places this time stamp in a packet and wakes up a data acquisition thread. Finally it acknowledges the interrupt and re-arms it prior to exiting.

To process the interrupts as efficiently as possible, it is necessary for the interrupt handler to directly communicate with the DAQ card without going through another module. The base I/O address and interrupt level of the DAQ card are necessary to do this. Protobox developed and tested a routine that obtains the base I/O address and interrupt level of the DAQ card. This routine is called in the initialization section of the ESTEEM module. The initialization section also creates the FIFO used to communicate between the real-time threads and the nonreal-time user process. Finally the initialization section creates the real-time threads. The initialization section is invoked when the module is first loaded.

The real-time threads perform the actual data collection after being awakened by the interrupt handler. The current ESTEEM version contains two of these threads. These threads read an A/D channel and place the data and the appropriate time stamp into the FIFO. This action wakes the user-mode process that is waiting on the FIFO. After completing this action the real-time threads suspend themselves waiting for the next wakeup event from the interrupt handler.

The termination section of the module is responsible for cleanup. It destroys the FIFO, destroys the real-time threads, and disarms the counter interrupts. This section is invoked when the module is removed/unloaded from the operating system.

**User-mode Process Software Development**

Protobox developed a prototype user-mode process to work in conjunction with the kernel-mode data collection processes. The user-mode process reads the data from the FIFO and takes appropriate action with the data. The current version of the user process reads the FIFO, determines which real-time thread placed the data into the FIFO, and then writes the time stamp out to a file designated for each counter.

## Performance measurement

Protobox measured the core interrupt handler performance under various conditions to assure that it provides deterministic performance. This interrupt handler is key to making the ESTEEM system provide accurate results under any simulation measurement condition. True real-time performance is imperative for a data collection system like ESTEEM. The software previously described represents the basic real-time core of ESTEEM. All future versions of ESTEEM will be built upon this core; therefore it is important that the performance of this core be accurately measured. To this end Protobox created the following setup to measure real-time performance (Figure 15).

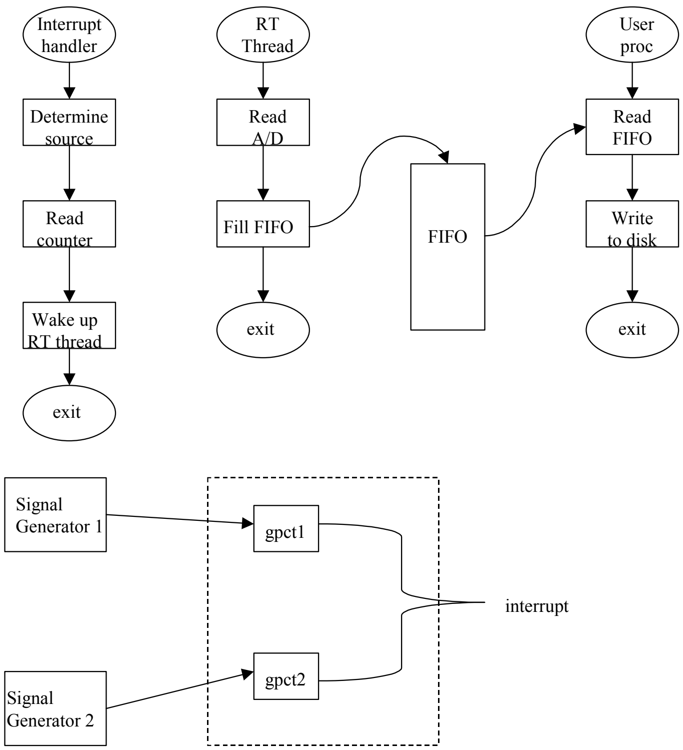



Figure 15. Deterministic performance test setup

Two signal generators were used for the asynchronous test. One was set to about 65 Hz, the other to about 64 Hz. For the synchronous test both interrupts were connected to the same source. A user process allows the user to select either external interrupt or both. The counters were set up to count at a 20 MHz rate and to reset to zero when the interrupt occurred. The interrupt handler reads the counter and places the counter value in a packet, which is eventually put into the FIFO by a real-time thread. The user process reads the time stamp from the FIFO and writes it out to a file. These data files were then imported into EXCEL for analysis. The counter value read incorporates the interrupt service time and the software overhead associated with reading the counters.

**Asynchronous Interrupt Test - processor idle.**

For this test the 64 Hz interrupt signal was connected to interrupt 1 and the 65 Hz interrupt signal connected to interrupt 2. No other activity was happening on the system. The test ran for approximately two minutes during which it collected 7,142 samples for interrupt 1 and 7,258 samples for interrupt 2. Figure 16 shows the histogram for interrupt1 and Figure 17 shows the histogram for interrupt 2.

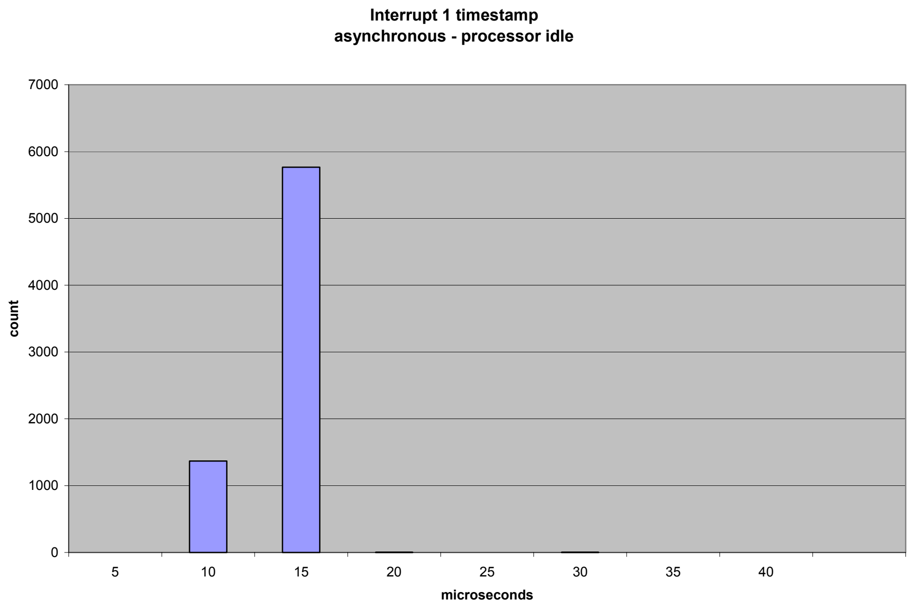



Figure 16. Asynchronous Interrupt Test – processor idle -- histogram for interrupt 1

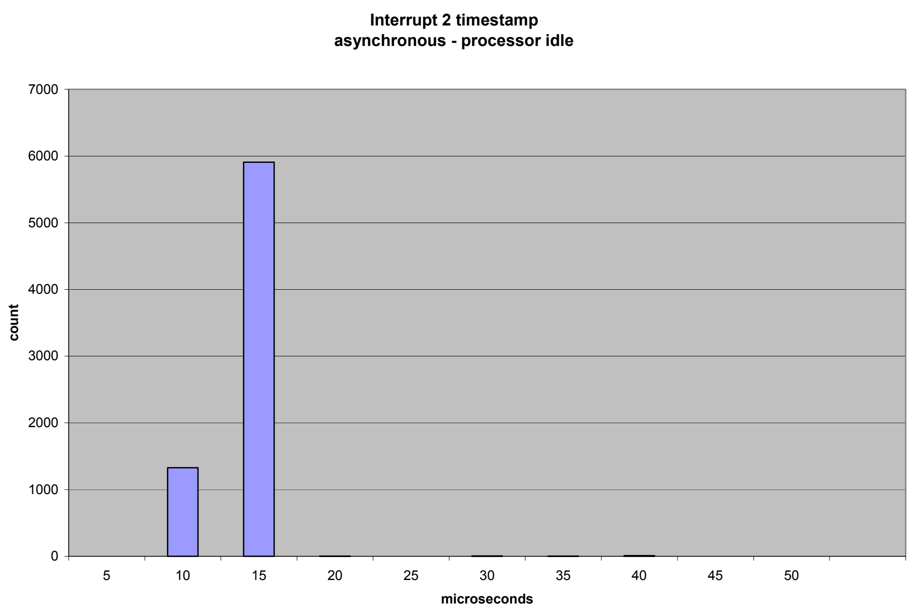



Figure 17. Asynchronous Interrupt Test – processor idle -- histogram for interrupt 2

As can be seen in the figures, the vast majority of samples fall in the 10 to 15 microsecond range.

**Asynchronous Interrupt Test - processor busy.**

This test used the same setup as the previous test except that the system was busy performing other activities. A recursive "grep" was performed starting in the root directory to cause constant disk accesses during the test run. A game called "snake race" was activated to cause graphics activity. The test ran for approximately two minutes during which it collected 7,148 samples for interrupt 1 and 7,251 samples for interrupt 2. Figure 18 shows the histogram for interrupt 1 and Figure 19 shows the histogram for interrupt 2.

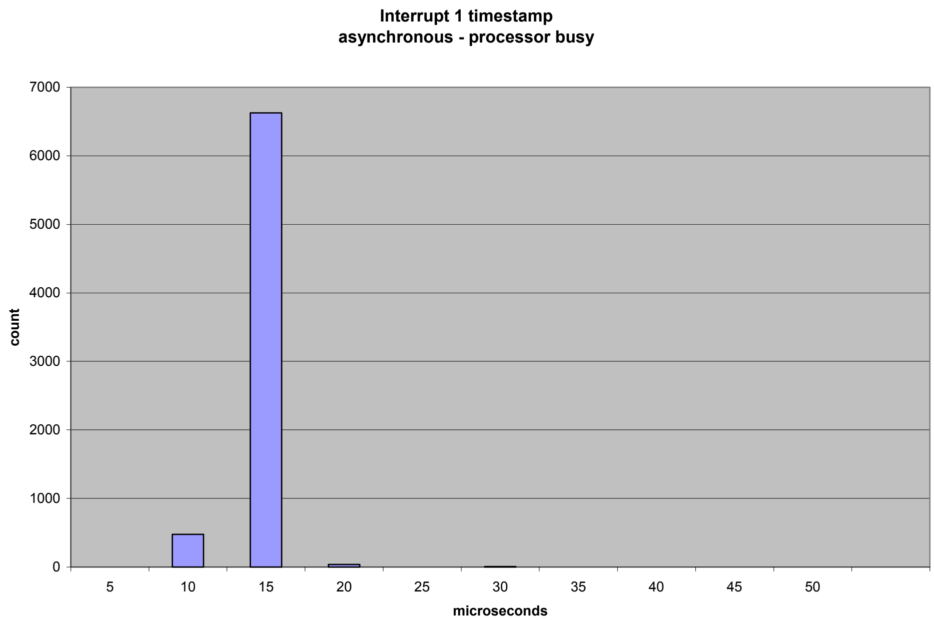



Figure 18.  Asynchronous Interrupt Test – processor busy -- histogram for interrupt 1

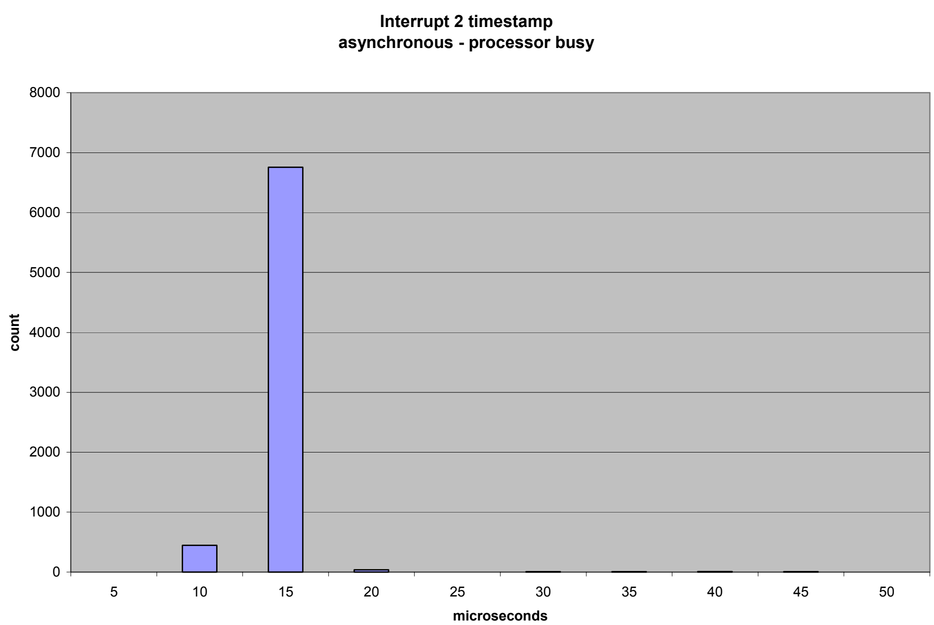



Figure 19.  Asynchronous Interrupt Test – processor busy -- histogram for interrupt 2

Even on a busy system the vast majority of samples fall within the 10 to 15 microsecond range.

**Synchronous Interrupt Test - processor idle.**

For this test both external interrupts were connected to the same source. This is a worst-case scenario. No other activity was happening on the system. The test ran for approximately two minutes in which it collected 7,202 samples for both interrupts. Figure 20 shows the histogram for interrupt 1 and Figure 21 shows the histogram for interrupt 2.

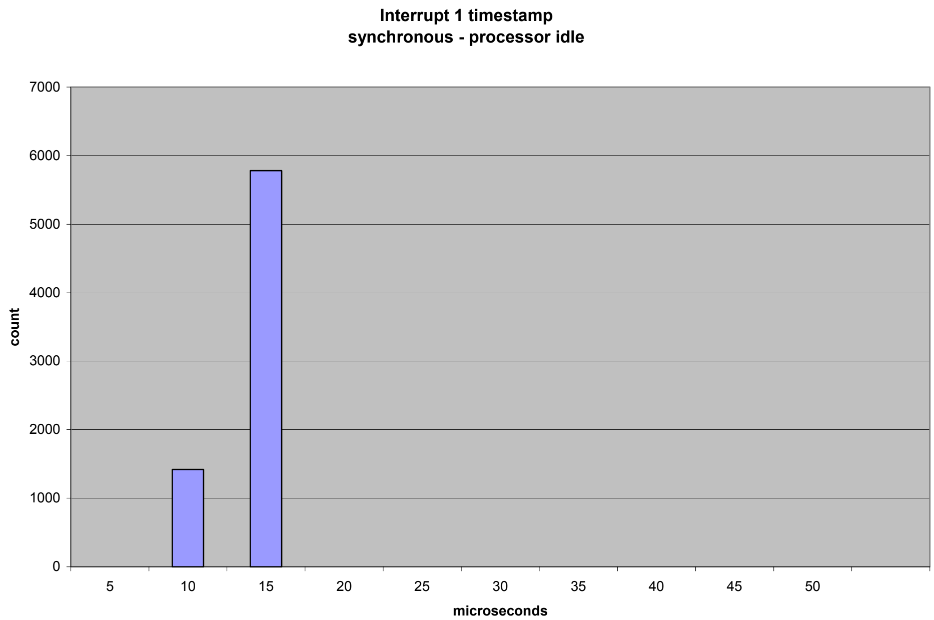



Figure 20. Synchronous Interrupt Test -- processor idle -- histogram for interrupt 1

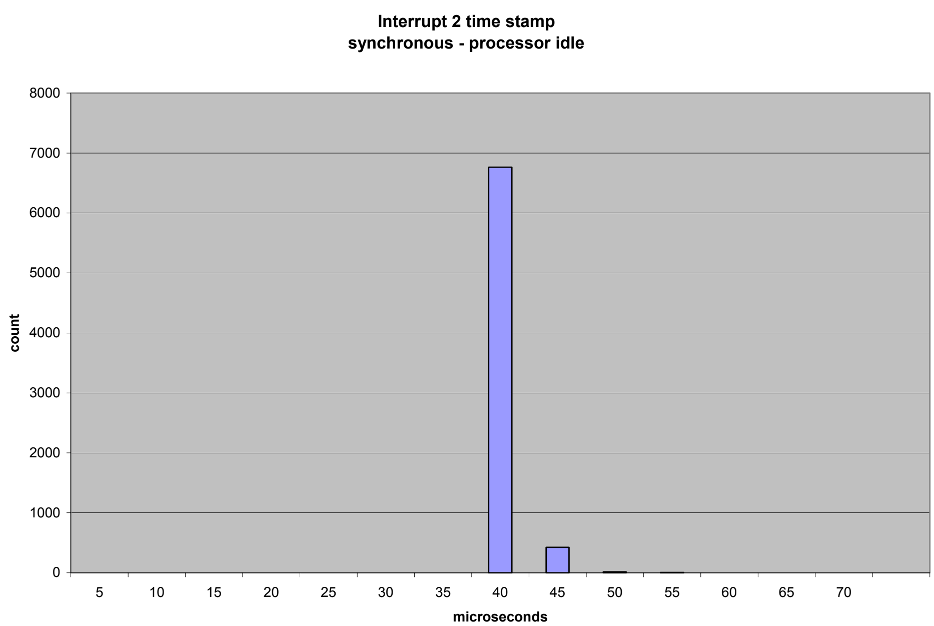



Figure 21. Synchronous Test -- processor idle -- histogram for interrupt 2

In this case it can be seen that interrupt 2 is being held off by interrupt 1. Even in this worst-case scenario the vast majority of values fall within the 40 to 45 microsecond range.

**Synchronous Interrupt test - processor busy.**

This test used the same setup as the previous test except that the system was busy performing other activities. A recursive “grep” was performed starting in the root directory to cause constant disk accesses during the test run. A game called “snake race” was activated to cause graphics activity. The test ran for approximately two minutes in which it collected 7,202 samples for interrupt 1 and interrupt 2. Figure 22 shows the histogram for interrupt 1 and Figure 23 shows the histogram for interrupt 2.

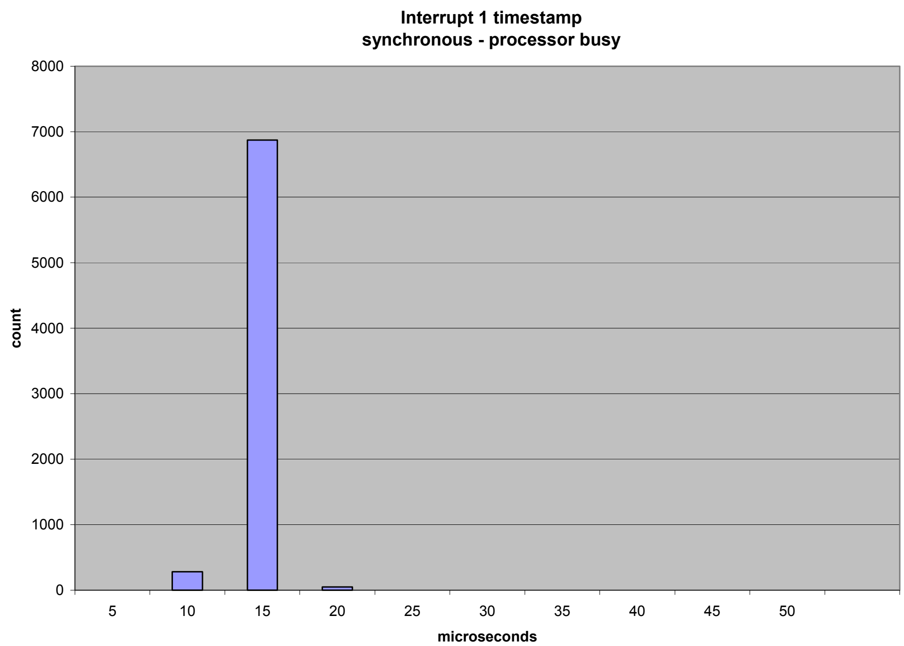



Figure 22. Synchronous interrupt -- processor busy -- histogram for interrupt 1

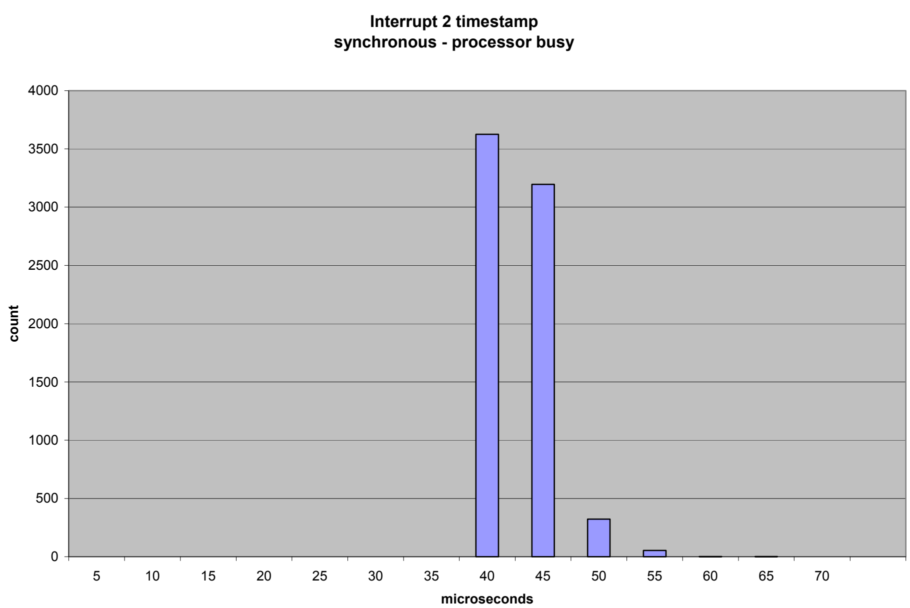



Figure 23. Synchronous interrupt test -- processor busy -- histogram for interrupt 2

Figure 24 summarizes results for the above testing. Protobox is extremely pleased with these results because they indicate that we will be able to achieve our goal of providing deterministic data timestamped to an accuracy better than 0.1 milliseconds between any two ESTEEMs located anywhere in the world. The maximum time to timestamp the data was 60.9 microseconds. Even with the system busy and interrupt 2 being held off by interrupt 1 the vast majority of timestamps fall within the 45 to 50 microsecond range.   All values are in microseconds.

| Timestamp | Async -idle | Async - busy | Sync -idle | Sync - busy |
|---|---|---|---|---|
| Interrupt 1 minimum | 4.7 | 8.9 | 9.7 | 9.8 |
| Interrupt 2 minimum | 9.7 | 9.8 | 38.1 | 38.3 |
| Interrupt 1 maximum | 32.6 | 32.3 | 15.8 | 18.2 |
| Interrupt 2 maximum | 40.9 | 41.6 | 53.2 | 60.9 |
| Interrupt 1 average | 10.3 | 10.9 | 10.3 | 11.1 |
| Interrupt 2 average | 10.4 | 10.9 | 39.1 | 40.9 |
| Interrupt 1 mode | 10.4 | 10.5 | 10.3 | 10.5 |
| Interrupt 2 mode | 10.3 | 10.5 | 38.9 | 39.5 |

Values in microseconds.

Figure 24.  Summary of software interrupt testing

**Interrupt stability**

Throughout testing, the interrupt pulse and the processing of the interrupt were monitored to ensure that no unexpected latency or variations occurred.  Figure 25 shows a copy of an oscilloscope used during the monitoring.  The top trace is in interrupt pulse.  The computer was set up to interrupt on the trailing edge, i.e., the low-to-high transition of the interrupt.  The timestamp occurred at the leading edge of the bottom trace.  Interrupt processing occurred during the "high" portion of the lower trace.  The data acquisition thread was initiated at the end of the lower pulse.  ESTEEM's goal is to service the interrupt within 100 microseconds (0.1 ms) of the interrupt.  For all testing that was done, the interrupt was always serviced within this time period.

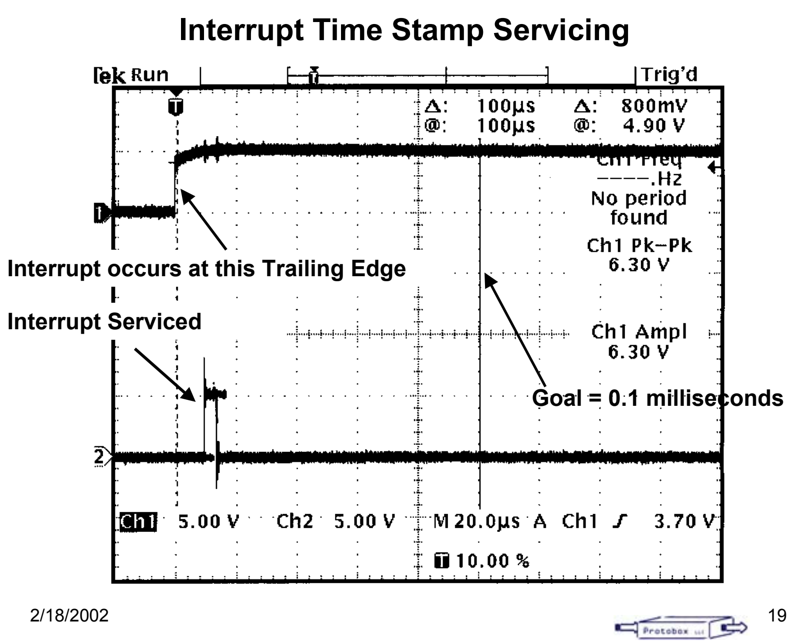



Figure 25.    Interrupt Time Stamp Servicing

Figure 26 indicates the stability of the interrupt servicing.  In this figure, the time scale has been expanded to 4.00 microseconds/cm.  The storage mode on the oscilloscope has been activated to capture the jitter.  As in the previous figure, the leading edge of the lower trace indicates when the timestamp occurred.

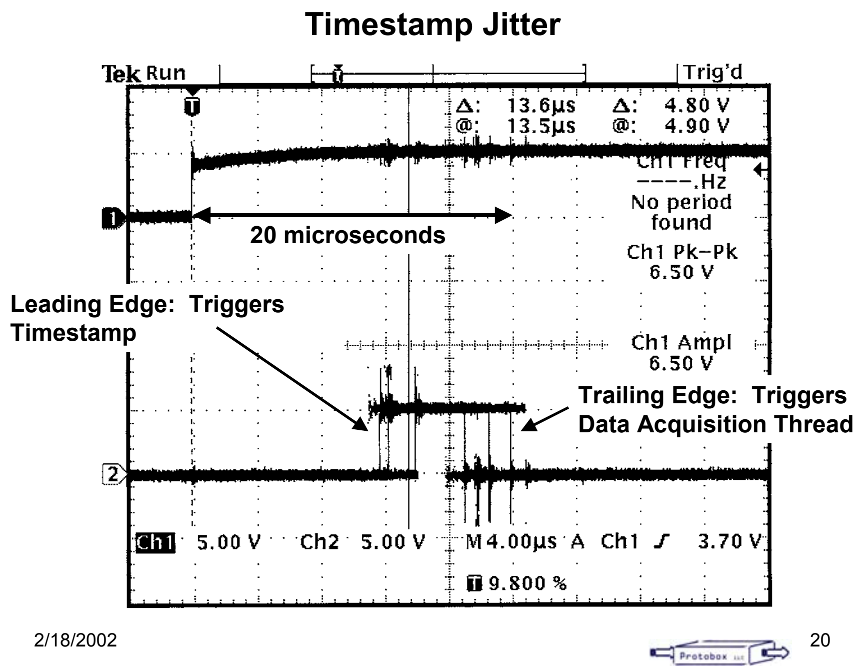



Figure 26. Timestamp jitter

## Human factors

Protobox LLC considers the human-factors issues related to the ESTEEM measurement system to be very important. In the past, data-gathering devices have concentrated on the capability to gather data, but the experiments and types of data gathered have been limited to engineering parameters. One of the goals of the ESTEEM project is to define test techniques and parameters that can provide the researcher with the type of data that is needed to determine the quality of the simulation on a Distributed Mission Training (DMT) network.

Several meetings were held between Protobox LLC and Dr. Jennie Gallimore of Humanwise Inc. throughougt the Phase I effort. Humanwise supported Protobox's Phase I development. Dr. Gallimore (a) met with AFRL/HEA representatives at Mesa, AZ to discuss human factors requirements, (b) conducted a literature search of the simulator testing subject, (c) defined a human factors pilot test and evaluation which could be used in Phase II, and (d) provided the concept and initial GUI examples for future development.

Dr. Gallimore also traveled to AFRL/HEA in Mesa, AZ on September 7, 2001. She discussed measurement requirements with Capt. Jeremy Hendrix and Dr. Winston Bennett. As a result of her visit, she had a better understanding of HEA's requirements. AFRL/HEA provided Dr. Gallimore with a "parameter list" that describes parameters of interest that are often needed.

**GUI Interface**

Dr. Gallimore supported Protobox LLC's development of GUI screens for control and interface to the ESTEEM system. She is responsible for ensuring that human factors aspects are considered during GUI development. A few of the screens are illustrated below.

ESTEEM GUIs fall into one of these general classes:

Set-Up and Configuration
Run-time
Post-run

The Set-Up and Configuration GUIs will permit the ESTEEM user to specify those specific signals, ranges, and labels that ESTEEM will monitor, such as stick roll and pitch, throttle position, switch states (discretes), etc. Individual Set-ups will be able to be saved and retrieved for later use. ESTEEM will collect the data during experiments and store the data is user-specified file names.

The Run-time GUIs will allow the ESTEEM user to view data on the display screen while it is being collected. This data will properly correlate data collected from ESTEEM units located throughout the network. Under normal operation this data will appear only a few seconds after the data was collected. ESTEEM waits to transmit the data back to the user until there is a lull in network traffic. During critical experiments, ESTEEM waits until the experiment is complete, then sends the data back to the user. This is done so that ESTEEM adds zero traffic to the network during a critical experiment. The user can change the appearance of the display screens without impacting the data collected. Some examples of such Run-time GUIs follow below.

The Post-run GUIs will provide a means for the user to retrieve and review data that was recently collected and stored in a temporary buffer. The Post-run GUIs allow the user to view data that was previously collected and stored as files. The Post-run GUIs also allow the user to export such data to third party packages, such as Excel and MatLab.

Run-time GUI examples

Figures 27 through 30 illustrate candidate Run-time GUIs, formulated during Phase I; these GUIs will serve as the baseline for the Phase II Usability Analysis and related refinement. Protobox LLC continues to work with Humanwise to define these GUI interfaces and their capabilities.

Figure 27 depicts the four-cockpit Viper simulation facility as connected to two other remote simulation sites. "Green" indicates network latency performance is within limits, and "Yellow" indicates that problems are occurring and thus, bear watching. The user can set the thresholds for the performance limits. If a failure occurs, such as unacceptable continuous latency between two nodes in a network, then the appropriate lines and nodes would become "Red."

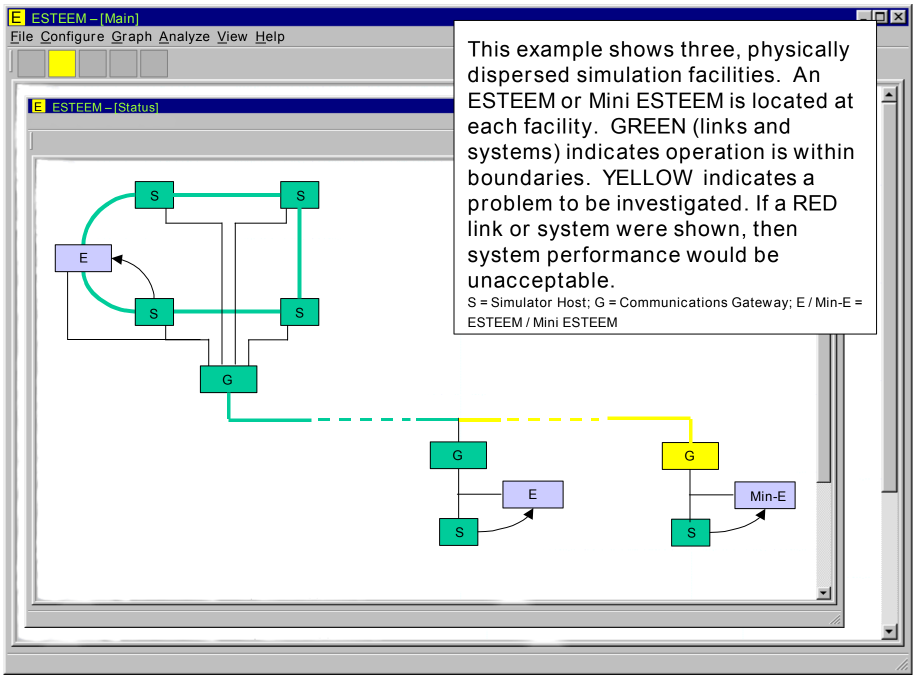



Figure 27. ESTEEM status GUI

Figure 28 illustrates a specific set of signals. Here we see the average latency between two simulators displayed. The yellow and red lines represent user-specified caution and warning thresholds for the latency.

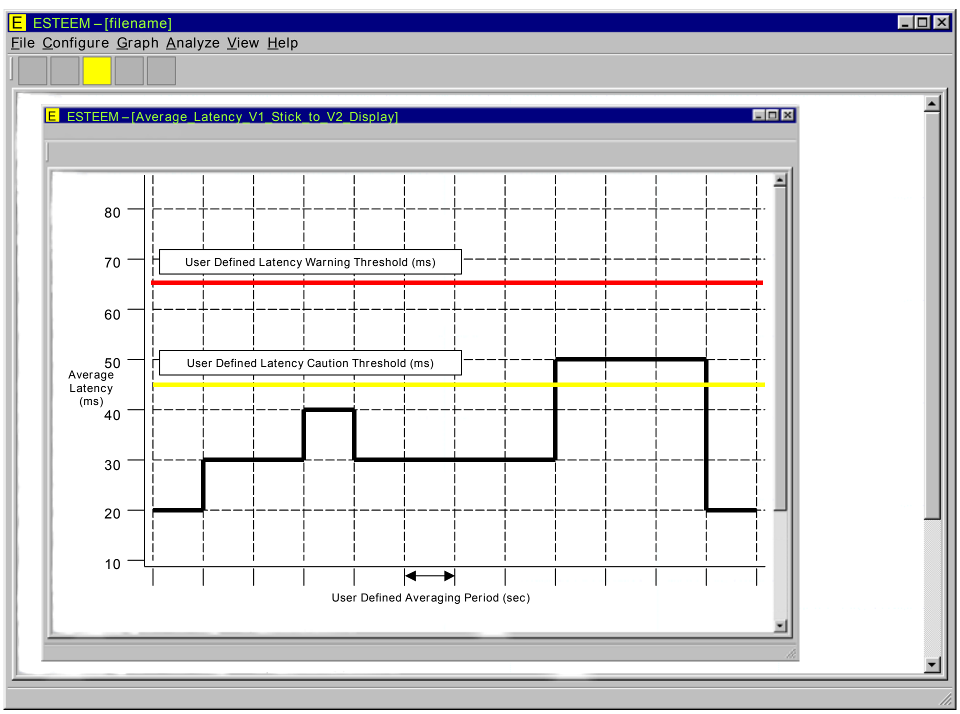



Figure 28. Signal latency with threshold limits

Figure 29 illustrates the state of user specified switches, in this case, four hypothetical switches in the Viper 2 cockpit.

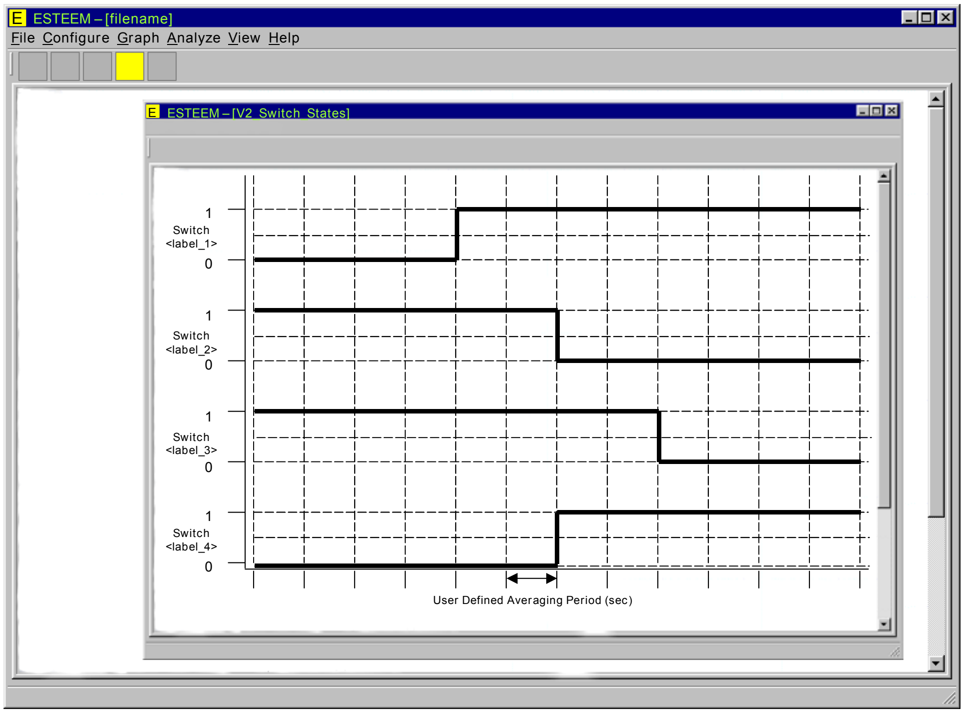



Figure 29. Discrete signal display

Figure 30 illustrates the roll response of the Viper 2 aircraft (i.e., the output of the V2 aircraft model based upon a specific stimulus, as defined below).

In this case:

(A) is the normalized value of the stimulus and response;

(B) is the time of occurrence (as timestamped by the ESTEEM timing subsystem);

(C) is the stimulus, a positive half-height step followed by two full height steps (negative and positive);

(D) is the roll rate response; and

(E) is the roll response.

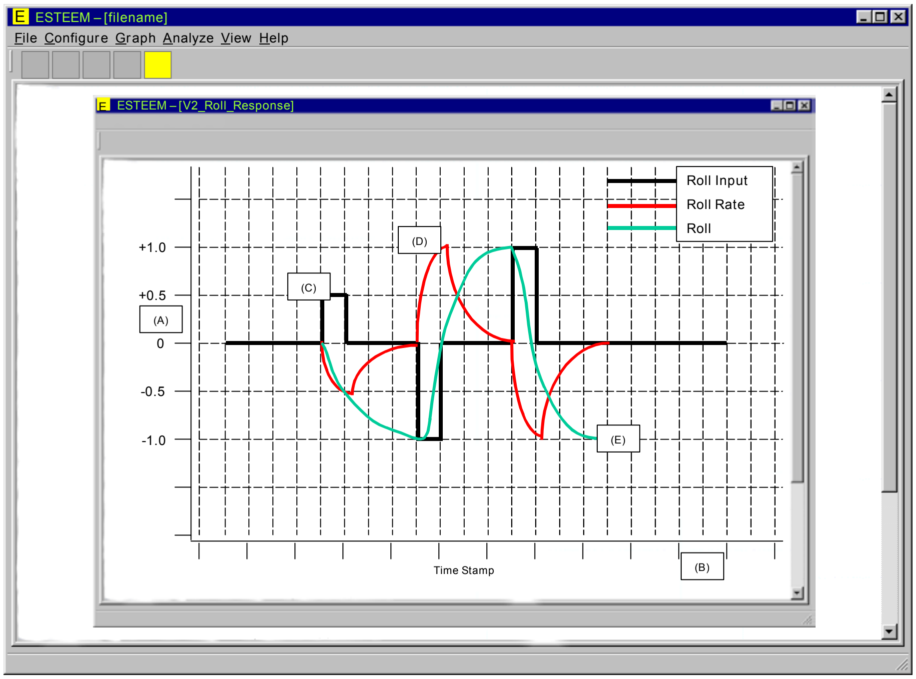



Figure 30. Typical plot of related signals

## Concepts for future ESTEEM development

Phase I of the ESTEEM project developed many of the key concepts and prototyped them in the demonstration system that was delivered to AFRL/HEA. Some additional concepts that require development to make the system into a usable tool are discussed below.

### GPS integration

The GPS will be integrated into the system so that data with time tags received at one simulation site can be accurately correlated with data at other sites. This integration will require installing a True Time GPS system into the ESTEEM system and developing the time-stamping software. The important part of this software is that the time-stamp associated with the data is read at the same time relative to the start of the routine. Testing will be done to verify that the amount of latency introduced by this routine is as small as possible, and that it is deterministic across systems. Phase I has developed the most deterministic interrupt handler ever used for time tagging network simulation data. Protobox will not do anything to degrade this response time. So far, the system that has been developed will be far more accurate and deterministic than any previously developed simulator performance measurement system.

### Reflective Memory Interface

To be compatible with simulator systems used at AFRL/HEA, a VMIC Reflective Memory Interface will need to be added to the ESTEEM system. This reflective memory interface will allow ESTEEM to gain access to AFRL/HEA's Viper host simulator variables such as aircraft state variables that are often important in analyzing the performance of the simulator. These variables are also helpful in tracing latencies through the simulator system. For example, end-to-end latency tests typically start by looking at the lateral stick input and end with the roll angle measurement of the video made by EVDAS. However, it is often important to know when the calculated roll angle changes within the host simulator. By having access to this state variable, the user can then determine how much of the lag is due to the aero calculations and how much of the latency is a result of the interface to the image generator and the image generator calculation time itself.

**HLA and DIS Services**

An HLA and DIS handler portion of the software will need to be developed so that the ESTEEM system can monitor simulator data packets as they are transmitted from one simulator to another. It must also be able to perform this task on the DMT network with the Portal concept that is being implemented by TRW. ESTEEM will be compatible with both the conventional single network simulation concept as well as with the Portal concept.

**Develop communication between ESTEEM systems**

A unique feature of ESTEEM is the ability to provide the operator with near real-time latency measurements during system operation. The Standard ESTEEMs and/or Mini-ESTEEMs must be able to communicate with each other to provide this capability. Communication software needs to be developed for this purpose. This software will employ an efficient custom protocol to support both transfer of commands and data. The communication software will be the top layer in a layered architecture that will support communications across both LANS and the DMT network.

**Develop communication across DMT Portals**

The Standard ESTEEMs / Mini-ESTEEMs must be able to communicate over the DMT network during network performance measurements. A review of the preliminary DMT portal design documents indicates that the portal will be capable of supporting other protocols besides HLA and DIS. Among these potential additional protocols are FTP, HTTP, SNMP, and HTML. These additional protocols, as well as the DIS protocol, need to be evaluated for their use as a "middle layer" to support ESTEEM communications over the DMT network.

**ESTEEM Security Issues**

Future ESTEEM implementations need to have the capability to be used within secure facilities. Many of the facilities that ESTEEM is likely to be used in will have security requirements. Full implementations of ESTEEM will be designed so that it can be completely declassified. The current plan is to include a 3.5" removable hard drive as the only drive in the system. This hard drive will serve to hold all of the Operating System, Application Software, as well as data files. This drive will remain unclassified until it is taken into a secure facility and connected to the simulation network. At that time the hard drive will become classified. The classified disk can then be retained in the facility safe or destroyed using proper procedures at the conclusion of the test. This technique is typically accepted by security organizations as a common-sense approach to assure that the system can be declassified and removed from a classified facility.

## Delivery and Demonstration

Protobox LLC delivered the ESTEEM Phase I prototype to AFRL/HEA during a visit on February 12, 2002. The system included hardware and software that was developed during Phase I as discussed in this report. A demonstration was provided to show some of the key features of the system. The system was capable of capturing data from four analog sources while simultaneously capturing Ethernet packets. Test signal generators were used to create the analog signals. A separate Linux machine was used to generate test Ethernet packets. Figure 31 illustrates the demonstration that was presented to AFRL/HEA representatives.

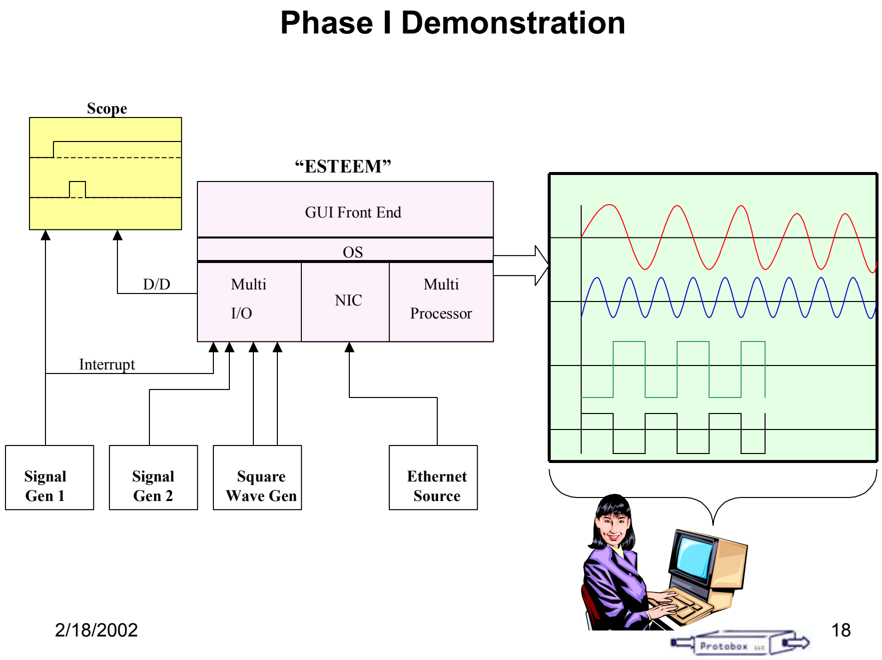



Figure 31. Phase I demonstration.

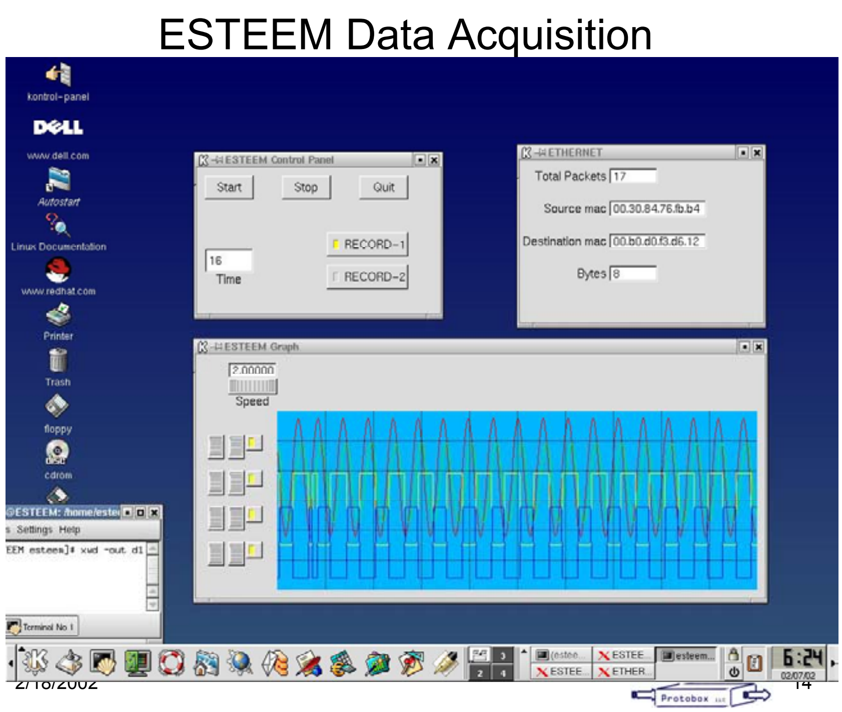
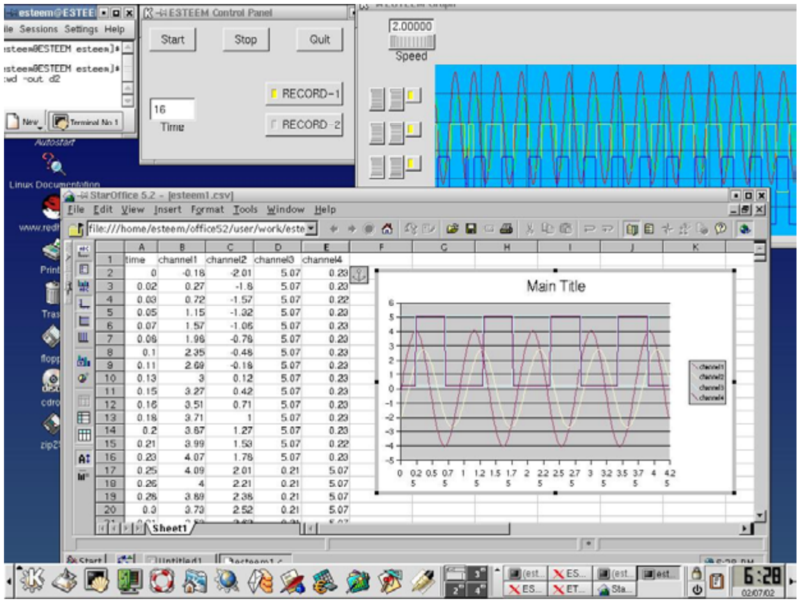



Figure 32. Data acquisition demonstration GUI

## Data Analysis




2/18/2002 15

Figure 33. Data analysis.

The Phase I demonstration included recording of sample data and analysis of the data using a "StarOffice" spreadsheet program. The data is saved in a tab-delimited format so that it can just as easily be imported into a standard Microsoft Excel spreadsheet.

### Summary

Protobox LLC believes that the ESTEEM system can make a significant contribution during the development of the DMT network. The Air Force has invested heavily in the DMT network and will continue to do so for a number of years to come. ESTEEM will provide a valuable tool to baseline the accuracy and latency of the DMT network during development. It will also serve as a verification tool to detect degradation in performance of interactive entities across the DMT network. For example, it is likely that when the DMT network is initially put into operation, there will be a good bit of spare bandwidth available, and that the performance of the network will be good. As additional entities and features are added to the DMT over a period of years, and as the percentage of available network bandwidth dwindles, it will be necessary to verify that the performance of the DMT network. ESTEEM is the ideal tool for the job. ESTEEM can be left continually on the network and periodically used by researchers to conduct experiments and to verify that the simulation systems are operating properly.

ESTEEM can also be used to pinpoint the location of performance problems. If an excessive interactive latency is noted during an experiment, ESTEEM can be configured to record additional points of interest between the two entities. Additional experiments can be conducted to break the overall simulation latency into its component parts. By dividing the latency into its components, researchers can determine where excessive latency is being introduced and concentrate their development efforts to improve those problem areas.

## Glossary

| | |
|---|---|
| AFRL/HEA | Air Force Research Laboratory/Human Effectiveness Directorate |
| DIS | Distributed Interactive Simulation |
| DMT | Distributed Mission Training |
| ESTEEM | Embedded Simulator Test Evaluation Monitor |
| EVDAS | Electronic Visual Display Attitude Sensor |
| GNOME | Tool used to build windows in Linux |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| HLA | High Level Architecture |
| Linux | ESTEEM's Operating System |
| LMCO | Lockheed-Martin Company |
| Min-E | Mini-ESTEEM |
| MTC | Mission Training Center |
| Portal | TRW's term for network interface unit/translator located at each simulation site. |
| RTI | Run Time Infrastructure |
| SNAP | Simulator Network Analysis Project |
| SOW | Statement Of Work |
| UI | User's Interface |
| Viper-n | One of the four fighter simulator cockpits at AFRL/HEA, Mesa, AZ |
| HumanWise | Name of human factors subcontractor's company |

## Bibliography

The references listed below were used to support Phase I R&D:


[Anderson 95] Don Anderson and Tom Shanley, Pentium Processor System Architecture, 1995.

[Anderson 99] Don Anderson and Tom Shanley, PCI System Architecture, 4th Edition, 1999.

[Barabanov 00] Michael Barabanov and Victor Yodaiken, Introducing Real-Time Linux; Linux Journal, 16 February 2000, http://www2.linuxjournal.com/cgi-bin/frames.pl/lj-issues/issue34/0232.html

[Boff 88] Kenneth R. Boff and J.E. Lincoln, J.E., Engineering Data Compendium: Human Perception and Performance. AAMRL, Wright Patterson AFB, Ohio, 1988.

[Bovet 01] Daniel P. Bovet and Marco Cesati, Understanding the Linux Kernal, 2001.

[Bryant 94] Richard Barry Bryant, Capt. D. Scott Douglas, Ronald Ewart, Gary Jeff Slutz, Dynamic Latency Measurement Using The Simulator Network Analysis Project; Proceedings of the 16th I/ITSEC, 28 November - 1 December, 1994.

[Bryson 90] S. Bryson and Fisher, S.S., Defining, modeling and measuring system lag in virtual environments. Stereoscopic Displays and Applications, SPIE vol 1256, 98-109, 1990.

[Bryson 93] S. Bryson, Effects of lag and frame rate on various tracking tasks. In Stereoscopic Displays and Applications, SPIE, Vol 1915, pp.155-166, 1993.

[CADIS 93] Communication Architecture for Distributed Interactive Simulation (CADIS), IST-CR-93-13, June 1993.

[Caro 98] Dick Caro, What does real-time mean anyway?, Industrial Computing Online, 1 October 1998, (http://www.isa.org/journals/ic/ accessed 2 October 2001).

[Cherepov 99] Mike Cherepov and Chris Jones, Hard Real-Time with RTX on Windows NT; Proceedings of the 3rd USENIX Windows NT Symposium, 12-13 july 1999, Seattle, WA.

[Cockayne 88] William T. Cockayne and Michael Zyda, Mobile Agents; Manning Publications, 1988.

[Cox 96] Andy Cox, et. al., Time Synchronization Experiments, Paper 96-14-091, presented at the 14th DIS Workshop,March 1996, Orlando, FL.

[Ellis 99] S. R. Ellis, et.al., Sensor Spatial Distortion, Visual Latency, and Update Rate Effects on 3D Tracking in Virtual Environments. In Proceedings Virtual Reality Annual International Symposium, March, 1999 Houston, TX., pp 218-221.

[ESTEEM 01a] Ron Ewart, Jeff Slutz, and Jennie Gallimore, *An Embedded Simulator Test Evaluation Monitor (ESTEEM) to Improve Distributed Mission training (DMT),* Interim Technical Report #1, 21 August 2001.

[ESTEEM 01b] Ron Ewart, Jeff Slutz, and Jennie Gallimore, *An Embedded Simulator Test Evaluation Monitor (ESTEEM) to Improve Distributed Mission training (DMT),* Interim Technical Report #2, 7 September 2001.

[ESTEEM 01c] Ron Ewart, Jeff Slutz, and Jennie Gallimore, *An Embedded Simulator Test Evaluation Monitor (ESTEEM) to Improve Distributed Mission training (DMT),* Interim Technical Report #3, 7 Nov 2001.

[Frank 88] L.H. Frank, et. al., Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. Human Factors, 30, 201-217, 1988.

[Furuichi 97] Masakazu Furuichi, et. al., Performance Evaluation Model of HLA-RTI and Evaluation Result of eRTI, presented at the 1997 Fall Simulation Interoperability Workshop, September 1997, Orlando, FL.

[Gum 87] Don R. Gum and Edward A. Martin, Ph.D., The Flight Simulator Time Delay Problem, presented at the AIAA Simulation Technology Conference, 17-19 August 1987, Monterey CA. Paper 87-2369-CP.

[Held 91] R. Held and Durlach, N., Telepresence, time delay and adaptation. In: Ellis, S.R. ed. Pictorial communication in virtual and real environments. New York: Taylor and Francis, 1991.

[Knight 01] Pamela Knight, et.al., Independent Throughput and Latency Benchmarking for the Evaluation of RTI Implementations, paper 01F-SIW-033, presented at the Software Interoperability Workshop, September 2001, Orlando, FL.

[Kuhl 00] Dr. Frederick Kuhl, , Dr. Richard Weatherly, Dr. Judith Dahmann, Creating Computer Simulation Systems – *An introduction to the High Level Architecture;* 2000, ISBN: 0-13-022511-8.

[Levinson 79] William H. Levinson, et. al., Effects of simulator delays on performance and learning in a roll-axis tracking task. Proceedings of the Fifteenth Annual Conference on Manual Control (AFFDL-TR-793134). Wright Patterson Air Force Base, OH: Air Force Flight Dynamics Laboratory. (DTIC No. ADA080563), 1979.

[McKee 97] Dr. Larry McKee, Latency and Its Effects on the Fidelity of Air-to-Air Missile T&E Using Advanced Distributed Simulations," presented at the SISO Fall Simulation Interoperability Workshop, 8-12 September, 1997, Orlando, FL.

[McLean 01] Thom McClean, Hard Real-time Simulation Using HLA, paper 01F-SIW-093, presented at the Software Interoperability Workshop, September 2001, Orlando, FL.

[McMillan 90] Grant R. McMillam, Ph.D., Cue Integration and Synchronization, Armstrong Aerospace Medical Research Laboratory, 8 January 1990.

[Monson 97] Steven L. Monson, et. al., Latency – The Adversary of Real-Time Distributed Simulation, Proceedings of I/ITSEC 1997.

[Park 99] K.S. Park and R. V. Kenyon, Effects of network characteristics on human performance in a collaborative environment. Proceedings Virtual Reality Annual International Symposium, March, 1999. p 104-111.

[Pitts 99] David Pitts, et. al., Red Hat Linux 6 Unleashed, 1999.

[Purdy 96] Lt. Stephen G. Purdy, Jr., et. al., A Comparison of Training Simulator Network Architectures, Wright Laboratory Control Integration and Assessment Branch, Paper release number ASC 96-2549.

[Purdy 98] Lt. Stephen G. Purdy, Jr., and Roger D. Wuerfel, A Comparison of HLA and DIS Real-Time Performance, Proceeding of I/ITSEC 1998.

[Red Hat 01A] Red Hat Linux 7.1, Getting Started Guide, 2001.

[Red Hat 01B] Red Hat Linux 7.1, x86 Installation Guide, 2001.

[Rubini 01] Alessandro Rubini and Jonathan Corbet, Linux Device Drivers, 2001.

[Slutz 96] G. Jeff Slutz and C.A. Brecht, Time Delay Study of the PIO Simulation, Halifax Corp. report under contract F33601-93-D-J016, 8 March 1996.

[Slutz 92] G. Jeff Slutz and Ron Ewart, An Electronic Visual Display Attitude Sensor (EVDAS) for Analysis of Flight Simulator Delays, AIAA paper, 1992.

[So 92] So, R.H.Y. and Griffin, M.J., Compensating lags in head-coupled displays using head position. Journal of Aircraft, 29, 1064-1068., 1992.

[So 94] So, R.H.Y., An investigation of the effects of lags on motion sickness with a head-coupled visual display. In: Proc. United Kingdom Informal Group Meeting on Human Response to Vibration, Institute of Naval Medicine, Alverstoke, 1994.

[So 95] So, R.H.Y., Optimizing the response of head-coupled systems to dynamic head movements. PhD thesis, Institute of Sound and Vibration Research, University of Southampton, 1995.

[So 95a] So, R.H.Y. and Griffin, M.J., Effects of lags on human-operator transfer functions with head-coupled systems. Aviation, Space and Environmental Medicine, 66, 550-556, 1995.

[So 95b] So, R.H.Y. and Griffin, M.J., Modeling the effects of display lags and reticle lags on head tracking performance. ISVR Contract Report No. CR 95/25, University of Southampton, 1995.

[So 95c] So, R.H.Y. and Griffin, M.J., Head-coupled virtual environment with display lag. In: Carr, K. and England, R., eds. Simulated and Virtual Realities - Elements of Perception. London: Taylor and Francis, 1995.

[So 96] So, R.H.Y. and Griffin, M.J., Use of phase-lead filters to compensate lags in head-coupled visual displays. IEEE Transactions on Systems, Man and Cybernetics, 1996.

[Solomon 98] David A. Solomon, Inside Windows NT; Microsoft Press, 1998.

[Swain 95] Steven D. Swain and T. F. Marz, DIS at Nine G's, Paper 95-13-042, presented at the 13th DIS Workshop, September 1995, Albuquerque, NM.

[Taylor 95] Darrin Taylor, Sc.D. and John Morrison, A DIS Variant for Reducing Latency in Long Haul Networks, MaK Technologies, Inc. Final Report for Contract F33615-94-C-3601, 29 January 1995.

[Taylor 96] Darrin Taylor, Sc.D, DIS-Lite & Query Protocol: Message Structures, presented at the 14th DIS Workshop, March 1996, Orlando, FL.

[Tharp 92] G. Tharp, G. et.al., Timing considerations of head-mounted display performance. Human Vision, Visual Processing and Digital Display III, SPIE vol 1666, 570-576, 1992.

[Timmerman ??] Dr. Martin Timmerman, Windows NT as Real-Time OS, pp. ?? to ?? in Dedicated Systems Magazine; <month, year>.

[TRW 01] Position Paper for Standard Portal Interface Specification, Version 2.0 for Distributed Mission Training Operations and Integration, DAL No. 67647-145; Prepared for the Department of the Air Force Aeronautical Systems Center by TRW; 12 June 01.

[TRW 00] Standard Portal Interface Specification, Version 1 for Distributed Mission Training Operations and Integration (DMT O&I), DAL No. 67647-132; Prepared for the Department of the Air force Aeronautical Systems Center by TRW; 20 Dec 2000.

[Valentino 98] George J. Valentino, et. al., An Agents Toolkit to Support Distributed Simulation; Proceedings of the 1998 Spring Software Interoperability Workshop, March 1998, Paper 98S-SIW-009.

[Valentino 96] George J. Valentino, et. al., Techniques for Reducing Latency in Distributed Flight Simulation, paper 96-3542, presented at the AIAA Flight Simulation Technologies Conference, 31 July 1996, San Diego, CA.

[Valentino 97] George J. Valentino, et. al., SNAPpy – The Windows NT, SMP, and VR-Link Based Low-Latency NIU and Latency Measurement System, Paper F97-SIW-043, presented at the 1997 Fall Simulation Interoperability Workshop, September 1997, Orlando, FL.

[Welsh 99] Matt Welsh, et.al., Running Linux, 3rd Edition, 1999.

[Wioka 95] M.W. Wioka, Lag in multi-processor virtual reality. Presence, 4, 50-63, 1995.

[Woodruff 86] R.R. Woodruff, et.al., Comparison of helmet-mounted visual displays for flight simulation. Displays, 7, 179-185, 1986.

[Woodyard 96] John Woodyard and Lt. D. J. Barnhart, Analysis of the Latencies in a Flight Simulator using the Simulator Network Analysis Project (SNAP), presented at the 14th DIS Workshop, March 1996.

[Yodaiken 99] Victor Yodaiken, RTLinux Manifesto, Proceedings of the 5th Linux Expo, March 1999.

[Zhao 01] Hui Zhao and Nicolas D. Georganas, Enabling Technologies for Real-Time Distributed Simulation, paper 01F-SIW-022, presented at the Software Interoperability Workshop, September 2001, Orlando, FL.

[Zhao 01A] Hui Zhao and Nicolas D. Georganas, Architecture Proposal of Real-Time RTI, paper 01F-SIW-025, presented at the Software Interoperability Workshop, September 2001, Orlando, FL.

www.labs.redhat.com/gug/users-guide/index.html GNOME 1.4 User's Guide

www.redhat.com Red Hat Linux Site