A PROTOTYPE OF A SIMULATION NETWORK USING THE DISTRIBUTED INTERACTIVE SIMULATION NETWORK STANDARD

Grace C. Mak-Cheng
Kenneth Doris
Grumman Corporation
Great River, New York

Frank Zinghini Jr.
Applied Visions, Inc.
Centerport, New York

ABSTRACT

With the emergence of technical standards for networking defense simulations, a means to evaluate the applicability of the new DIS PDU's and the draft DIS networking requirements to joint services applications is needed. This paper describes a rapid prototyping testbed which was developed to network two simulating devices using a fiber distributed data interface (FDDI) local area network (LAN). A Distributed Interactive Simulation Interface Unit (DIU) was developed to interface each simulating device to the FDDI LAN. The intent of the DIU was to off-load the host processor and to minimize the changes to existing trainers as a result of networking. The DIU performed the dead reckoning and translation of the data from(to) the DIS PDU protocol to the host computer format. The DIU was developed using both Ada and C and the simulating devices were developed in Pascal.

The DIS PDU protocol was also used as the communication protocol between the simulating device and the DIU. Thus, this prototype implemented the DIS PDU protocol in three different languages. The "lesson learned" from this DIS implementation and suggestions to undeveloped areas, such as Simulation Management and Electronic Warfare, in the DIS standard are discussed.

ABOUT THE AUTHORS

Grace Mak-Cheng is a Software Technical Specialist in Grumman's training systems organization. She is currently responsible for an Independent Research and Development project involving real-time networking of distributed simulation. She actively participates on the Standards for the Interoperability of Defense Simulations. Ms. Mak-Cheng holds a Masters in Computer Science from New York Institute of Technology and a Bachelor of Electrical Engineering from New York Polytechnic Institute of Technology. She is an expert in the use of structured analysis and structured design software methodologies especially as applied to embedded software systems.

Kenneth Doris is a Technical Advisor in Grumman's training systems organization. He is currently responsible for managing the Independent Research and Development projects, including the IRAD project involving real-time networking of distributed simulation, for Grumman's training systems. He *actively participates on the Standards for the Interoperability of Defense Simulations.* Mr. Doris holds a Bachelor of Electrical Engineering from Rensselaer Polytechnic Institute and has twenty-five years experience in Computer Image Generation and C^3I.

Frank Zinghini is the founder and president of Applied Visions Inc., a software development and consulting company specializing in simulation, networking, and graphical computing. Mr. Zinghini holds a Masters in Computer Science and Bachelor of Electrical Engineering from the State University of New York at Stony Brook.

A PROTOTYPE OF A SIMULATION NETWORK USING THE DISTRIBUTED INTERACTIVE SIMULATION NETWORK STANDARD

Grace C. Mak-Cheng
Kenneth Doris
Grumman Corporation
Great River, New York

Frank Zinghini Jr.
Applied Visions, Inc.
Centerport, New York

INTRODUCTION

The need to network trainers to provide simulated battle environment tactical training is more pressing now than ever as a result of cuts in the defense budget, as well as environmental concerns. However, trainers which currently exist are diverse in many ways. They differ in fidelity, run on different computer systems, are written with different computer languages and they each have their own database. Recent advances in technology make possible the networking of existing, remotely located trainers and simulators having varied architecture and levels of complexity. This permits each of the trainers and simulators to participate as members of a tactical force within a coordinated tactical environment.

The Army has successfully networked simulators for tank, helicopter, Bradley fighting vehicles, surface-to-air missile systems, and generic A-10 attack aircraft using the SIMNET protocols. For the most part, these devices run on homogeneous computer systems.

The basic concepts of Distributed Interactive Simulation (DIS) are an extension of the Simulation Networking (SIMNET) program, especially in the format and content of the messages to be exchanged. These messages, called protocol data units (PDUs), have been modified in DIS to accommodate a completely heterogeneous mixture of networked systems. Another significant change from SIMNET to DIS is in their underlying communications software. SIMNET is based on a set of protocols developed specifically for that project; DIS uses "off-the-shelf" commercially available protocols.

One of the missing elements in the DIS development process is the availability of systems to test out the evolving specification. This paper describes a rapid prototyping testbed which was developed at Grumman for that purpose. The testbed includes the use of a fiber distributed data interface (FDDI) local area network (LAN) coupled with commercial TCP/IP and UDP/IP communications packages. A Distributed Interactive Simulation Interface Unit (DIU) was developed to interface each of our simulating devices to the FDDI LAN. The DIU, similar in concept to the Cell Adapter Unit concept of ADST, performs dead reckoning, DIS translation and other DIS networking related tasks. In addition simulation and network management functions are performed by a Simulation Coordination Console (SCC) software package which may be hosted on any DIU in the network.

DISTRIBUTED INTERACTIVE SIMULATION (DIS) ISSUES

DIS PDU ISSUES

The DIS PDU standard has been submitted to the Institute of Electrical and Electronics Engineering (IEEE) to become an IEEE standard. The PDUs currently defined as part of the standard (versus those *recommended* by the standard) are primarily intended to support simulation of conflicts between entities within visual range of one another. PDUs which support beyond visual range (BVR) conflicts fall into the recommended class and are still in a state of flux. Examples of such PDUs are those which deal with electromagnetic, infrared, and acoustic parameters. Since these issues have little or no back-

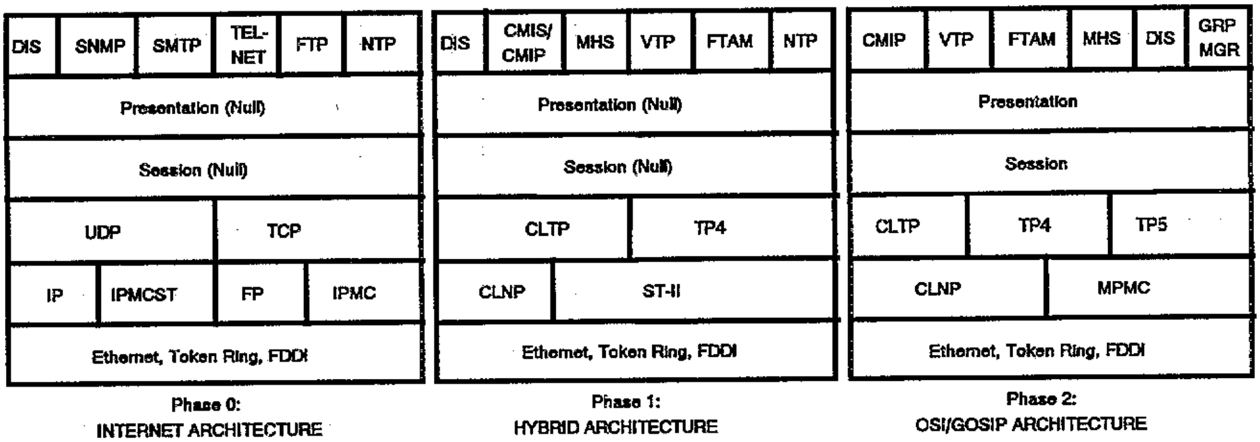

Figure 1 DIS Three Phase Communication Architecure

ground in SIMNET, it is imperative that they be explored thoroughly as part of the DIS development process. One of the near term goals of the testbed is to explore the ability of this class of PDUs to support simulation of BVR conflicts. Similarly, the ability to support the simulation of environmental effects does not yet exist in the DIS specification, but is planned as a future enhancement. Here again the testbed will be used as a tool to evaluate the emerging portions of the specification.

NETWORK COMMUNICATION ISSUES

The communication between the individual simulators in a network involves three key areas, namely network architecture, software protocols, and transmission medium. A draft version of the Communication Architecture for Distributed Interactive Simulation (CADIS) was issued in March of 1992 by the Institute for Simulation and Training (IST). This document contains the standards that specify the communication architecture, the recommended practices for implementing the communication architecture, and the rational to support these standards and practices. The goal of the DIS communication architecture is to be compliant with the OSI/GOSIP architecture. Since GOSIP is still under development, DIS will use standard, commercially available communication protocols in the interim. The strategy involves the three phases shown in Figure 1.

The performance of the DIS communication architecture involves analyzing the DIS bandwidth requirements and latency requirements. Figure 2 shows the bandwidth analysis which was included in the CADIS document for three sample DIS exercises. The network traffic analysis shows traffic in terms of bits/second, PDUs/second, and packets/second. Figure 3 shows the DIS latency requirements as stated in the CADIS document.

NETWORK MANAGEMENT ISSUES

In the current form of the DIS standard, management functions that would allow Network Management, Simulation Management and Performance Measures do not exist. The Simulation Management subgroup of the Interface-Time/Mission Critical subgroup of the DIS standardization committee is currently designing a protocol which will address these management functions. The SCC developed as part of this project is intended to explore these issues.

DISTRIBUTED INTERACTIVE SIMULATION PROTOTYPE TESTBED

The purpose of the discussion up to this point has been to inform the audience of some of the issues currently being addressed by the DIS community. What follows is a description of an on-going independent research and development (IR&D) program at Grumman aimed at exploring critical DIS-related issues.

TABLE NOTATION:

A = NUMBER OF ARTICULATED PART RECORD
H = NUMBER OF ARTICULATED PARTS HIT
E = NUMBER OF EMITTERS
B = NUMBER OF BEAMS PER EMITTER
T = NUMBER OF TARGETS PER BEAM

PDU	PDU NAME	FORMULA
ESPDU	ENTITY STATE PDU	1152+128A
FPDU	FIRE PDU	704
DPDU	DETONATION PDU	800+128H
EPDU	EMITTER PDU	192+E(160+B(304+96T))

SAMPLE PDU SIZING

	A	*H*	*E*	*B*	*T*	*ESPDU*	*FPDU*	*DPDU*	*EPDU*
TANK	5	1	1	1	1	2220	1132	1356	1180
AIRCRAFT	20	2	3	1	2	4140	1132	1484	2588
SURFACE SHIP	50	5	10	1	5	7552	1132	1868	10060
						OVERHEAD BITS/PDU =			428

SAMPLE RATES PER ENTITY TYPE PER PDU TYPE

	LOW RATE				*HIGH RATE*			
	ESPDU	*FPDU*	*DPDU*	*EPDU*	*ESPDU*	*FPDU*	*DPDU*	*EPDU*
TANK	0.2	0	0	0.2	2	0.1	0.1	1
AIRCRAFT	0.2	0	0	0.2	8	0.1	0.1	4
SURFACE SHIP	0.2	0	0	0.2	1	0.1	0.1	2

SAMPLE EXERCISE TRAFFIC ESTIMATES

% ENTITIES AT HIGH RATE	0%	20%	40%	60%	80%	100%
% ENTITIES AT LOW RATE	100%	80%	60%	40%	20%	0%
SAMPLE EXERCISE #1						
600 TANKS	408,000	1,030,656	1,653,312	2,275,968	2,898,624	3,521,280
100 AIRCRAFTS	134,560	982,320	1,830,080	2,677,840	3,525,600	4,373,360
10 TACTICAL VOICE LINKS	650,000	650,000	650,000	650,000	650,000	650,000
TOTAL TRAFFIC (BITS/SEC)	*1,192,560*	*2,662,976*	*4,133,392*	*5,603,808*	*7,074,224*	*8,544,640*
(PDUs/SEC)	*600*	*1172*	*1744*	*2316*	*2888*	*3460*
(PACKETS/SEC)	*81*	*187*	*293*	*399*	*505*	*610*
SAMPLE EXERCISE #2						
24 SHIPS	84,538	201,896	319,254	436,612	553,970	671,328
50 AIRCRAFTS	67,280	491,160	915,040	1,338,920	1,762,800	2,186,680
3 TACTICAL VOICE LINKS	237,500	237,500	237,500	237,500	237,500	237,500
TOTAL TRAFFIC (BITS/SEC)	*389,318*	*930,556*	*1,471,794*	*2,013,032*	*2,554,270*	*3,095,508*
(PDUs/SEC)	*54*	*185*	*316*	*448*	*579*	*711*
(PACKETS/SEC)	*32*	*74*	*115*	*157*	*199*	*241*
SAMPLE EXERCISE #3						
200 TANKS	136,000	343,552	551,104	758,656	966,208	1,173,760
100 AIRCRAFTS	134,560	982,320	1,830,080	2,677,840	3,525,600	4,373,360
5 TACTICAL VOICE LINKS	325,000	325,000	325,000	325,000	325,000	325,000
3 TACTICAL DATA LINKS	237,500	237,500	237,500	237,500	237,500	237,500
TOTAL TRAFFIC (BITS/SEC)	*833,060*	*1,888,372*	*2,943,684*	*3,998,996*	*5,054,308*	*6,109,620*
(PDUs/SEC)	*304*	*652*	*1000*	*1348*	*1696*	*2044*
(PACKETS/SEC)	*61*	*139*	*217*	*296*	*374*	*452*

Figure 2 DIS Bandwidth Analysis with Bits, PDUs, and Packets/Sec

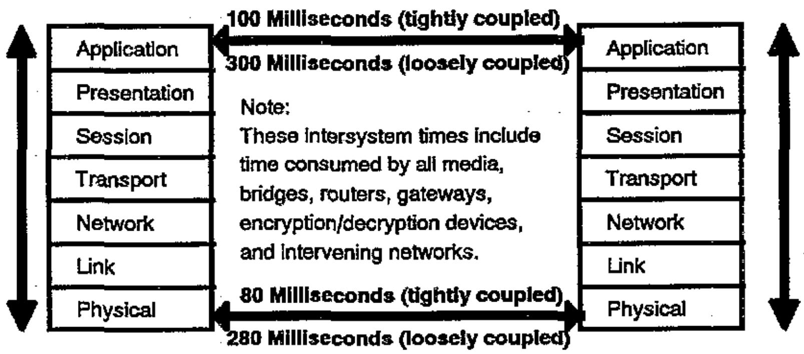

Figure 3 DIS Standard Latency Values

DIS INTERFACE UNIT (DIU)

The purpose of the DIS Interface Unit (DIU) is twofold:

- To ease the integration of existing trainers into DIS networks by minimizing their internal modifications and relieving them of the additional processing burden of dealing with the DIS network and DIS functions.
- To standardize the manner in which newly developed trainers tie into DIS networks

In our prototype testbed, each simulating device (represented for now by PCs) is interfaced to the DIU via an Ethernet connection. Each DIU is then linked together over a FDDI LAN. One of the primary purposes of our testbed is to evaluate the DIS PDUs in a beyond visual range application. Thus, one of our PCs simulates three entities, an E-2 aircraft, a F-14 aircraft and a Bear D aircraft. The other PC simulates a second F-14 aircraft. Figure 4 illustrates the prototype testbed. Each PC provides the operator with a number of display and control screens, all implemented within the Microsoft Windows environment. The displays include unclassified versions of E-2C and F-14D radar presentations with the data on each screen being driven as a direct result of the DIS link traffic.

Software hosted on the DIU performs the following functions:

- Dead reckoning on the targets between updates (this includes entity state and EW data).
- Coordinate transformation of the entity's location from its trainer's representation (e.g. geodetic) to the DIS World Coordinate representation (i.e. geocentric).
- Data filtering, so that only data of interest to the associated trainer is sent to it.
- Collision detection of entities simulated by the associated trainer with other entities in the DIS exercise.
- Translating data between the DIS PDUs and internal host computer formats.
- Handling the communication protocols for local area network communications.

The testbed DIUs are implemented on Concurrent Computer Corporation 7500 computer systems. The hardware for this system includes:

- Two Motorola 68040 CPUs running at 25 MHz
- A LANCE Ethernet Controller
- A FDDI Network Interface
- A GA1500/1550 Graphics Subsystem which includes SP-X11 (X Windows System Software) with SP-XMotif (OSF/MOTIF User-Interface Software)

The operating system that runs on the Concurrent 7500 machines is called RTU (real time UNIX) and is compatible with the AT&T System V.3 to the Base System and Kernel Extension definitions as validated by the System V Verification Suite (SVVS). The Base System definition describes the services that define the runtime behavior of the operation system as seen by an application program. The Kernel Extension provides additional operating system services. In addition, components of other

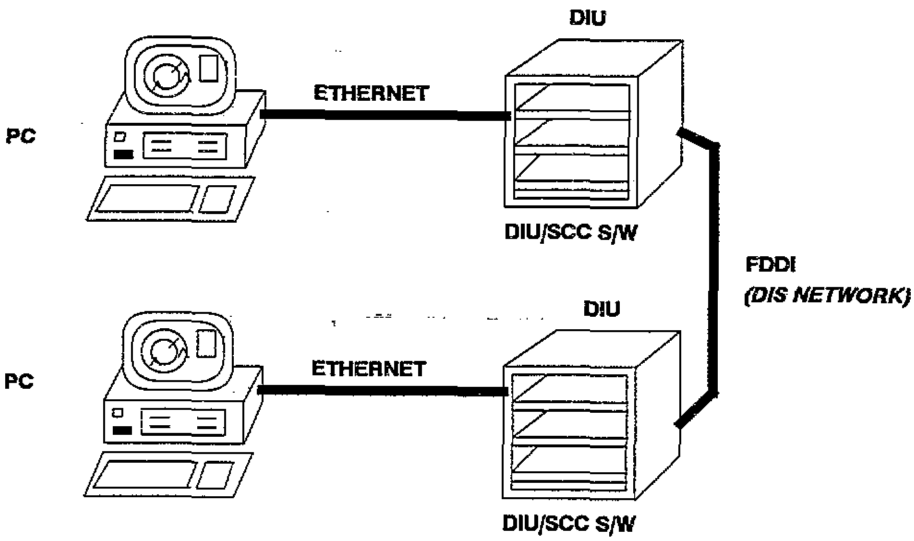

Figure 4 DIU Rapid Prototype Test Bed

extensions and of Berkeley Software Distribution (BSD) 4.2 UNIX are present in RTU. Threads, which is a unique feature to the RTU system, provide an efficient processing facility for RTU applications.

The DIU code was written in C and Ada. The multitask requirements of the DIU were implemented using multiprocessing, the standard method of forking new processes on UNIX systems, and threads, the RTU enhancement to the UNIX system that lets a single process run multiple tasks that share memory and other resources.

DEAD RECKONING

The DIU maintains a dead reckoned model of the entities simulated by its associated trainer as well as dead reckoned models of other entities "of interest" in the exercise. The dead reckoning algorithm to be used for a particular entity is defined in the Entity State PDU for that entity. The dead reckoning algorithms were implemented in accordance with Appendix I of the DIS standards document. We are currently investigating the dead reckoning requirements for EW data based on the latest version of the Emissions PDU as defined by the Emissions Subgroup of the Interface-Time/Mission Critical Subgroup.

TCP/IP, UDP/IP PROTOCOLS AND FDDI LAN

As recommended in the CADIS document, the TCP (Transmission Control Protocol) and the UDP (User Datagram Protocol) were used as our interim network communication protocol. The TCP protocol (which is designed for point-to-point dialogue) was used for communications between the "host simulators" (the PCs in our testbed) and the DIU, and between the DIU and the SCC (the SCC will be discussed later in the Simulation Management section). The UDP protocol (which is designed for one-to-many broadcasting) was used for communications between the DIUs during a simulation exercise.

The TCP and UDP protocols are standard products available from a large number of vendors. This implementation used products from Concurrent's "RTnet-TCP Network Software" in the DIUs and "Distinct TCP/IP for Windows" in the PCs. The permissible transmission media for a DIS network include Ethernet, Token Ring, and FDDI. We have chosen to experiment with FDDI for several reasons. The use of FDDI, which has a tenfold increase in bandwidth over Ethernet, will mitigate the impact of the considerable overhead that GOSIP compliance infers. In addition, FDDI allows for more growth to the number of devices which can be tied to the network. FDDI has the capability of connecting 1000 physical connections on a total fiber path of 200 kilometers. Since FDDI uses a timed token rotation protocol, there is a guaranteed upper bounds on time between successive arrivals of the token at a station. The deterministic nature of a FDDI network will give a better quantitative evaluation of network performance. The added security advantage of FDDI is another consideration. Fiber optic cables are virtually impossible to tap, do not conduct electricity and are immune to outside interference, thus allowing for more reliable signal transmission. The drawback to FDDI at the present time is its cost, both in the form of controller boards as well as cabling. As with many other aspects of computer technology, it is expected that these costs will decrease rapidly in the near future.

SIMULATION AND NETWORK MANAGEMENT

The current DIS standard document does not define the Simulation and Network Management functions needed for DIS. In our prototype testbed, the operation and control of the network was established with the implementation of the Simulation Coordination Console (SCC). The SCC is a network control point and is implemented as an operator's console attached to any one of the DIUs. The SCC performs the following Simulation Management functions:

- Exercise Setup
- Exercise Start/Restart
- Exercise Freeze/End

The SCC is designed to move around to various nodes. Its operation involves broadcasting its address on the network at regular intervals to inform interested players of the exercise. Upon detecting an interested player, the SCC request and receives a list of entities which can be simulated by the player. The SCC operator can then decide if the player is allowed to participate in the exercise and what entities it can simulate. This method of simulation management will ensure that all the necessary players are on the network prior to the exercise start and also allow late players to enter the network during an exercise.

HUMAN INTERFACE

The Motif Graphical User Interface was used in the implementation of the SCC operator console and our Status displays. The Status displays were used to monitor activity on both the DIS FDDI network and the Simulator's Ethernet network. Motif was chosen since it is rapidly becoming the defacto industry standard for graphical displays.

The simulating devices on the PCs were designed to show the dynamics of the DIS system. Here the user interface was implemented as a task running under Microsoft Windows since Motif is not available on these platforms. Some of the features which were included were:

- the ability to change the heading of an entity
- the ability to fire missiles (i.e. the ability to dynamically create entities)
- the ability to obtain entity information (i.e. the entity type, its location, airspeed, etc.)

The DIS PDUs were also used as the communication protocol between the simulating devices and the DIU with some minor changes (e.g. the simulating device expected the entity's location in terms of geodetic coordinates).

LESSONS LEARNED

COMMUNICATIONS SOFTWARE

The benefit of using TCP/IP and UDP/IP lies in their universal interoperability across the various computer and workstation vendors. In many ways this is amazing when one considers that the UNIX community is divided into two camps (AT&T and Berkeley) with each camp having *several versions currently in use. We have found* the interoperability between the camps and versions to be real, but have reason to believe that the relative performance of different versions and implementations to be less consistent. What follows is a simplified version of one of the performance issues we explored - simplified by the fact that we spare the reader excruciating detail that only UNIX programmers would find interesting.

The Concurrent software provides access to TCP/IP and UDP/IP via both the AT&T "Transport Layer Interface" (TLI) and the Berkeley "socket" mechanism. They differ in the following ways:

- TLI is media- and protocol- independent. It allows applications to run over any transport protocol that supports the TLI interface. TLI is found only in later versions of AT&T UNIX.
- The sockets interface has historically been tied to the Internet protocol suite, TCP/IP and

UDP/IP. Sockets are supported in both AT&T and Berkeley UNIX, and, to the best of our knowledge, all extant versions of them.

Our first implementation of the DIU communication used the socket interface since the code could easily be ported to computer systems supporting only a socket-based interface. However, after some preliminary benchmarks, we found that the performance of our implementation on the Concurrent 7500 systems were less than what was desired (in the order of 30 msec per packet). The communication code was then modified to use the TLI interface, resulting in an immediate order of magnitude increase in throughput. We subsequently proved in the lab that one machine could use the TLI interface to talk to another which used the socket interface, thus demonstrating their interoperability despite the performance differences. We are continuing to explore other possibilities for improving performance while maintaining interoperability.

LANGUAGE ISSUES

Our DIS PDUs were coded in several different languages at different stages of the project, including Ada, C, and Pascal. Each language presented us with its own set of advantages and disadvantages, but overall Ada was perhaps the best compromise. Our Ada implementation took advantage of the work performed by the University of Central Florida's Institute for Simulation and Training (UCF/IST). For a more detailed discussion of the application of various languages to the DIS protocols see the paper entitled "Journey to Babel..." in these proceedings.

NETWORK ISSUES

The packets which are sent between simulation participants are usually small, less than 250 bytes. Most network and network communication protocol benchmarks are obtained by measuring the time it takes to transmit a large file from one machine to another. Our preliminary benchmark show that performance improves with larger packet sizes (up to the maximum IP buffer size: 1500 bytes for Ethernet and 4352 bytes for FDDI). There has been some discussions in the DIS community on the concatenation of PDUs to improve performance.

The implementation of the communication protocols are usually propriety in nature; Silicon Graphics is one of the few vendors who actually supply their TCP and UDP source code. Vendors must be aware of the performance requirements of DIS. The RTnet-TCP software include network parameters which can be modified for performance tuning. The FDDI parameters can also be modified to enhance performance.

FUTURE DIRECTIONS

In the near future we plan to utilize the testbed to evaluate DIS's ability to support simulation of radio traffic between entities and to explore security issues. The radio traffic will include digitized voice and simulated data link traffic (e.g. Link-4A, Link-11, and JTIDS) between the simulated F-14s and E-2Cs. The security provisions will cover the use of public and private keys and various encryption techniques.

CONCLUSION

DIS is intended to serve as a tool for Combat Development, System Acquisition, Test and Evaluation and Training communities. The requirements needed to meet the needs of each of these communities are still being developed. The strategy of a phased, evolutionary approach for GOSIP compliance as defined in the CADIS document is worth noting. Those involved with developing the standards should investigate interim solutions so that a DIS environment can be implemented now. As implementers, our designs must allow for changes due to user needs, experimentation requirements, or changes in the DIS architecture as it evolves. Proper design techniques are needed in order to evaluate the connectivity issues required by DIS.