PERFORMANCE LIMITATIONS OF THE DIS INTERFACE

Rodney A. Long, Eric E. Anschuetz, & Lawrence R. Smith
Naval Air Warfare Center, Training Systems Division
Orlando, Florida

ABSTRACT

Distributed Interactive Simulation (DIS) Standards are being established to allow for connectivity and interoperability of dispersed simulations through the standardization of application layer protocols. However, the underlying datagram design is governed by the network bandwidth thus limiting what information can be shared between simulations. The finite bandwidth of serial networks limits how much information can be transferred from one point to another within a specified period. In addition, interfacing to a DIS environment requires a computational element capable of filtering information needed by the individual simulator and performing common functions necessary to interact in this distributed environment. Filtering of simulation data is required since most PDUs are transmitted using broadcast addressing. Dead reckoning provides an engineering tradeoff which reduces network bandwidth, but increases the computation necessary at the simulation interface.

Functions like filtering, dead reckoning, simulation management, collision detection, and time stamping are performed at the DIS interface. The time required to accomplish these functions as well as reliable Ethernet and FDDI communication for DIS is deterministic. The purpose of this paper is to identify the performance limitations of accomplishing the DIS interface as well as to identify the time required to perform the basic functions that make up the DIS interface.

ABOUT THE AUTHORS

Rodney A. Long is a computer engineer at Naval Air Warfare Center Training Systems Division (NAWCTSD) in Orlando, Florida. He graduated from the University of South Carolina with a B.S. in Computer Engineering. He currently works in the NAWCTSD DIS Technologies Group, focusing on the DIS interface area. He can be reached at NAWCTSD, 12350 Reasearch Parkway, Orlando, FL, 32826 or (407) 380-4110.

Eric E. Anschuetz is a computer engineer at NAWCTSD. He graduated from Eastern Michigan University with B.S. degrees in Computer Science and Mathematics. His current duties include software development for the DIS Network Interface Unit (NIU). He was also a participant in the 1993 I/ITSEC DIS demonstration.

Lawrence R. Smith is an electronics engineer in the NAWCTSD DIS Technologies Group. He received his B.S. in aerospace engineering at the U.S. Naval Academy in 1980 and his M.S. in aeronautical engineering at the Naval Postgraduate School in 1988.

PERFORMANCE LIMITATIONS OF THE DIS INTERFACE

Rodney A. Long, Eric E. Anschuetz, & Lawrence R. Smith
Naval Air Warfare Center, Training Systems Division
Orlando, Florida

INTRODUCTION

Distributed Interactive Simulation (DIS) Standards are being established to allow for the interoperability of disparate simulation systems through the standardization of application layer protocols. Through the use of the DIS communications architecture, individual systems can be interconnected to allow real-time, mutual interaction in a common synthetic environment. DIS protocols are used to transmit the minimum amount of information necessary to represent "ground truth." Using this data, each individual system is responsible for determining its own perception of the environment.

Unfortunately, today's technology contains certain bottlenecks that limit the amount of data that can be exchanged in real-time. These constraints include network capacity between systems, as well as the computational load on the individual systems. As a result, engineering tradeoffs are being made which reduce network bandwidth requirements, but increase the computational load at the simulation's network interface.

THE DIS NETWORK

The goal of DIS is to enable *distributed* simulations to interact in a common environment. Therefore, connectivity is a major consideration. Local Area Networks (LANs) can offer data rates from Ethernet's 10 Megabits per second (Mbps) to FDDI's 100 Mbps. However, in moving to a Wide Area Network (WAN), the data rates drop off dramatically due to delays in routers, switches, and the physical transmission medium. T1 networks are the most widely-installed WANs and offer a maximum data rate of only 1.54 Mbps.

Network Limits

Many factors influence DIS bandwidth during a networked exercise. These include: total number of entities, mixture of entity types, type of exercise or scenario, choice of dead reckoning algorithm (including positional and angular thresholds), and security requirements. Presently, the majority of network traffic involves Entity State Protocol Data Units (PDUs). [1]

An Entity State PDU provides specific information such as entity type, location, velocity, orientation and dead reckoning parameters. These PDUs are required to be sent at some minimum rate (e.g., every 5 seconds) by every entity and may also be sent much more frequently depending on entity dynamics.

A BBN review of DIS network loading from I/ITSEC-93 showed peak load periods which contained roughly 230 packets per second for 210 entities (generated by 49 applications). Most packets were about 186-200 bytes, including the overhead of UDP, IP and 802.3 header information. At the high end, this equates to approximately 0.37 Mbps. [2]

Filtering was performed by BBN to isolate various hosts. One host sent traffic from 8 live F-15s, accounting for about 20% of the network traffic; 96% Entity State PDUs, 2% Emission, 1% Transmitter, 1% other. As was expected, the fast movers (aircraft) produced the majority of network traffic, consisting mostly of Entity State PDUs.

Using an F-14 simulator from the NASNET program, we performed a variety of maneuvers to determine the worst case PDU production rate. With an update rate of 30 Hz, the system performed dead reckoning using DR algorithm #4 (second order position, first order orientation) with a tolerance of 1 meter for position and 3 degrees for orientation. Performing the "corkscrew," an ascending spiral, the system peaked at 15 Entity State PDUs/s. This is a typical flight pattern used after a bombing run to evade ground fire.

Network bandwidth saturation is a known problem which will affect very large exercises. Ethernet LANs have been observed to congest significantly at around 60% capacity; this is less than 20 times I/ITSEC-93 peak traffic. In comparison to major exercises being planned, I/ITSEC-93 was a rather small DIS exercise involving only a limited subset of PDU traffic. Other PDUs to be used in future exercise scenarios, such as the Emission or Signal PDUs, may have a more substantial impact on bandwidth than the Entity State PDU.

DIS NETWORK INTERFACE UNIT

Interfacing to a DIS environment requires a computational element capable of filtering information needed by the individual simulator and performing common functions necessary to interact in the simulated environment.
These functions are currently being defined by the DIS Interface Subgroup in the DIS Interface Functional Requirements Document (FRD).

The Naval Air Warfare Center Training Systems Division (NAWCTSD) has developed a DIS Network Interface Unit (NIU) as part of a Cooperative Research & Development Agreement with Motorola. The NIU performs several of the functions identified in the FRD including filtering of DIS Protocol Data Units (PDUs), dead reckoning, coordinate conversion, time stamp generation, and entity collision detection. These functions are described further in the following paragraphs.

DIS PDU Filtering

Filtering of simulation data is required since most PDUs are transmitted using broadcast addressing. The NIU will filter incoming PDUs to decrease the amount of data being processed. Filtering may be performed by exercise number, PDU type, entity kind, entity domain, entity category and distance from ownship.

Dead Reckoning

Dead reckoning is a method of position/orientation estimation used to reduce transmission of Entity State PDUs. By estimating the position and orientation of other systems' entities, it is not required for the application to receive a report about every change in position/orientation that occurs in the remote entities it is dead reckoning. An update is only required when a change in position/orientation differs by a certain amount from the dead reckoned position/orientation. Dead reckoning provides an engineering tradeoff which reduces network bandwidth, but increases the computation necessary at the simulation interface.

Coordinate Conversion

When existing simulation systems are upgraded for DIS, coordinate conversion is often necessary. The NIU will convert between the DIS standard World Coordinate System (Geocentric) and Topocentric, Universal Transverse Mercator (UTM), or Geodetic. Every incoming and outgoing PDU could undergo a coordinate transformation. The conversion process becomes a tradeoff between precision and computational load; precision is directly related to the amount of computations performed.

Time Stamp Generation

Time stamping is used to indicate the time at which the data in the PDU is valid. The time stamp represents units of time passed since the beginning of the current hour. The NIU uses relative time stamping.

Collision Detection

The NIU compares every local entity position with every remote entity position. When the NIU determines that the position of an outside entity is within a specified distance of any of the local targets, it will issue a Collision PDU.

CPU LOADING

The NIU was implemented as part of the NASNET F-14 simulator. The NIU runs on a Motorola 187 board using a Motorola 88100 RISC CPU (see Table 1 for benchmarks). For synchronization purposes, the NIU was run at 30 Hz, which is a common iteration rate for aircraft simulators.

Our first measurements determined that the major CPU intensive functions were dead reckoning and coordinate conversion. Loading effects due to time stamping, PDU filtering and collision detection were minimal in comparison. Upon further analysis, we realized that the Motorola 187 board did not perform transcendental functions such as sine and cosine in hardware. Since dead reckoning and coordinate conversion are math intensive, a hardware implementation of these functions would significantly improve performance.

SYSTEM	MIPS	SPEC INT 92	SPEC FP 92
88100 (33 Mhz)	50	27.7	18.8
* HP 735	76	52.4	149.8
* HP 750		48.1	75.0

TABLE 1: CPU PERFORMANCE BENCHMARKS

* Other computer systems shown for comparison.

The NIU performs dead reckoning using geocentric coordinates and then converts them to the coordinate system used by the simulation application. As this is a function of the number of entities in the Entity Table, we decided to test for the maximum number of entities that could be dead reckoned and had to undergo coordinateconversion. A Semi-Automated Force (SAF) program generated a variety of entities at a constant velocity, with DR algorithm #2 (first order position, no orientation). Using the UTM coordinate system, the NIU began missing frames at approximately 65 entities.

NIU MEMORY REQUIREMENTS

The NIU stores Entity State information in an Entity Table, which includes data from the Entity State PDU as well as other data such as the current dead reckoned position. The current implementation of the NIU can store approximately 1000 entities using 2 MB of memory. Since the CPU cannot process this amount of entities in real-time, memory does not become a factor.

RECOMMENDATIONS

As can be seen, the network interface can easily become overloaded. There are different approaches to solving the problem, none which are completely satisfactory.

The easiest approach is to wait for technology to build faster and cheaper computers. Given recent advancements in technology, this could be in the near future. However, it does not solve today's problems.

Using multicast addressing, the amount of traffic arriving at the network interface will be greatly reduced. Filtering will be done at the hardware level vice software. However, large exercises can be envisioned where a simulator's "area of concern" would include more entities than it could process.

A final alternative, is to filter the information that is processed for the simulation application. While a high level filter at the network is useful for filtering out network traffic from other exercises, a "world view" filter enables only data which is critical to the simulation application to be processed. Using this method, dead reckoning and coordinate conversion would be performed only on the most critical entities as defined by the application.

CONCLUSION

The Interface Subgroup is investigating the development of test tools and procedures to validate basic interface functions. Using functional definitions, the SubGroup is defining a process to: characterize basic parameters, evaluate interface functions, develop test requirements, validate interface performance, and document DIS interface products' capabilities and limitations.

REFERENCES

[1] Guidance Document (DRAFT), Communication Architecture for Distributed Simulation (CADIS), IST-CR-92-21, November 1992.

[2] Seeger, Joshua Dr. , "Network Oriented Scalability", DIS Workshop, March 1994.