NETWORKED SIMULATION

Wim G. de Jong
TNO Physics and Electronics Laboratory
The Hague, The Netherlands

ABSTRACT

Pioneered and guided by the US Defense Advanced Research Project Agency (DARPA) a major simulation technology development takes place. This technology is called Simulator Networking or Distributed Interactive Simulation (DIS) and integrates key simulator technology with computer communication technology. Networking many simulators over large distances is a promising concept, not only to train individuals and crews to act in an cooperative team but also to support the development of equipment by operating and evaluating real or simulated equipment within a simulated (virtual) environment. When used together these technologies create a system which provides a common "electronic playing field" on which simulators can interact in real-time.
The concept of networked simulation is based upon designing each simulation element (object, entity) as an autonomous entity, communicating its appearance (visual, electromagnetic, acoustical) through standardized messages via communication networks (LAN/WAN) to each other entity that is called upon the virtual playing field.
One of the many critical factors that needs further research when networking simulators is transport delay in relation with communication services.

Keywords: Networked simulation, simulator networking, distributed interactive simulation, simulator interoperability, advanced distributed simulation technology.

1. INTRODUCTION

Some years ago the networking of simulators got a push by a research and development project initiated by the Defense Advanced Research Project Agency. With assisting funding from the US Army a project called SIMulation NETworking (SIMNET) was started. The goal was to create a complete combat environment with manned combat vehicle simulators. These simulators were connected to the synthetic environment using both local and wide area networks.

The key technologies that contributed to the SIMNET project were:
- Distributed Computing
- Local and Wide Area Networks
- Computer Image Generators
- Computer Generated Forces (CGFs)

The cost of state-of-the-art computertechnology allowed each simulated vehicle sufficient computational power to process the data to simulate its own performance, while interacting with one another via local and wide area networks. Moderate cost computer image generators created adequate out-of-the-window visual scenes for the crews in the simulated vehicles. CGFs were populating the battlefield with friendly and enemy forces in addition to the vehicle simulators.

Realizing the potential benefits to be derived from networked simulators a process was initiated in 1989 to define a family of standards that would allow networking of both existing and new simulators and expanding the SIMNET concept. This expanded concept is called Distributed Interactive Simulation (DIS).

2. DIS BASIC CONCEPTS AND DESIGN PRINCIPLES

The purpose of DIS is to allow dissimilar simulators distributed over a large geographical area to interact in a team environment. Basic concepts and design principles are (Ref. 1):

2.1 Object Oriented Entity Design

Each simulation element will be designed as an autonomous entity. Individual entities will include a "public" and "private" component. Multiple entities will be connected through their public components to form simulation systems which represent virtual environments. The public component, designed as a separate module, handles the exchange of data between entities as well as any processing required to compensate for transmission delays and asynchronous arrival of data. For the purpose of discussion, the public component will include an entity state vector and a system state vector. The entity state vector maintains current values of the variables which describe the state of the entity. The system state vector maintains current values of variables which describe the state of conditions existing across the simulation system. While the public component must be "standard" across the system, the private component creates only the interactions and representations of the environment which are required for the simulation element created by the entity.
Figure 1 gives an architectural representation of a DIS entity.

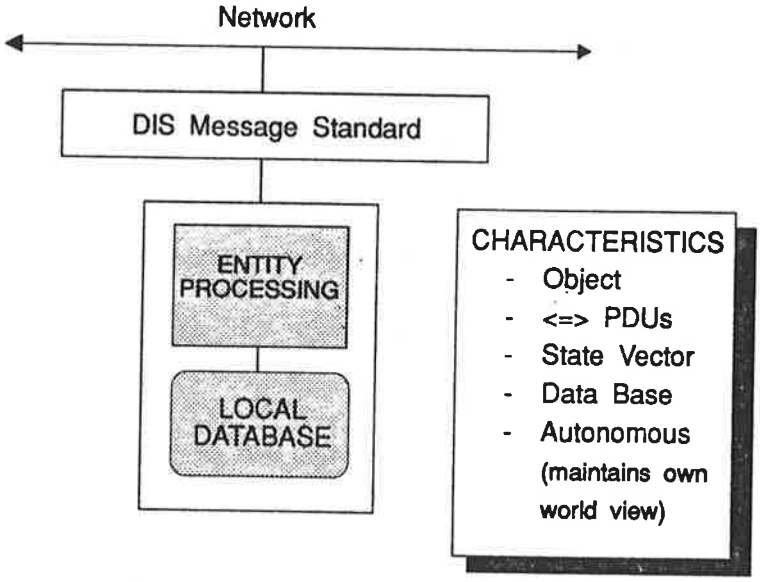


Figure 1: *DIS Entity*

2.2 Entity Sphere of Interaction - Cause and Effect

The private component of each entity will compute an active simulation region within a "sphere of interaction" i.e. for each entity the sphere of interaction defines the spatial region in which state vector data from other entities must be monitored and processed in order to maintain the interactive simulation within the private component of the entity. Effects on the simulated environment are caused by results of actions initiated by the individual entities. Results such as collisions may be computed by the entity which occur within the entity sphere of interaction and indicated as an event change in the state vector.
In other cases actions initiated by the entity such as active emissions may be continuously present over intervals of time and indicated as a state change in the state vector. In either case, the public component of the entity will transmit the change in its own state vector variables to all other entities affected by the change. Likewise, entities affected by these actions will compute the effects of the action and/or results received from the initiating entity and update its own state variable to show the change in state caused by the action. In this manner entities initiating actions will be responsible for notifying other entities in the system of the actions taken. Entities affected by the actions will be responsible for determining the effects of the action and notifying other entities in the system of resulting changes in entity state caused by the action.

2.3 Gaming Area - Environment Model Data Base

In order to maintain ground truth within the simulation system, each host computer must access a common representation of the environment (land, ocean, atmosphere and space). Hence, digital terrain data bases used by individual entities must, as a minimum, use the same "survey markers" as a common reference for generating terrain surfaces and overlay of cultural features and objects. Likewise, all host computers must have common representations of ocean, atmosphere and space environment models.

Given standard file structures for relating cultural features and object models to the survey markers, the extent to which terrain representation is identical within each host computer will depend upon the degree that the terrain generation formatting and rendering processing is the same within each host computer and image generator. Certainly, consistent representations of terrain and interactions can be accomplished within any given host computer. However, the cause and effect design requirements described above will require further consideration of how to describe and define correlation between entities which are using different formatting and rendering processes for terrain generation commonly referred to as different fidelity levels. These considerations for terrain representation apply equally as well to the ocean, atmosphere and space environment models in separate host computers.

2.4 Model Designs

Model designs and algorithms used within the individual host computers to create dynamic simulations of system performance, man-machine interactions and general representation of the environment must consider that data elements used in computing the models will, in part, be received from other entities in the system.

Moreover, the model designs should assure that variables or parameters which affect the model performance can be easily reset. Also, model designs should include provisions for system state variable parameters which affect the model's performance, e.g. atmospheric attenuation coefficient which changes as a function of climate and weather conditions. Again, concern over model fidelity used in the individual host computers will have to be adressed. In order to keep the system designs tractable it may be necessary to allow some limited number of discrete values for model and system state variables initially which may be extended as experience grows and need dictates.

2.5 Synchronous and Asynchronous Interconnections

Conventional centrally controlled simulations use time steps to synchronize the advancement of the simulation. In these cases, computations required to determine interactions between entities and changes in entity status are completed during a prescribed time interval; the simulation is updated to reflect these changes at the end of the time interval.

In the case of asynchronous interconnections such as those demonstrated by SIMNET, each entity updates state parameters and transmits the new values whenever the change in these parameters exceeds preset thresholds. Thus the update of parameters occurs asynchronously within the simulation system. To re-establish a synchronous simulation environment within individual entities, dead reckoning algorithms are used to extrapolate the state variable parameters of all external entities to the same current time of the individual entity. For reliable simulations, the extrapolating algorithms must be powerful enough to compensate for latency caused by transmission delays between entities and the lag in updating state variable changes.

2.6 System Management

Operation of a simulation system comprised of several individual entities interacting to form a virtual environment representation will require some design principles for system management. System Management (SM) will require the capability to initialize or "set" the values of the system state vector variables and all entity state vectors that will be connected in the system at the beginning of the exercise. Also SM will load model parameters and data bases in individual entities when required for customizing entity simulation performance to specific characteristics and defining the exercise gaming area. Likewise, SM will initialize communication services and interface parameters necessary for connecting the individual entities. During a simulation exercise, SM will be responsible for changes in the system state vector, updates or modifications of data bases which apply between entities, addition and deletion of the host computers connected to the system, supervising data collection taskings and any other activity which applies to multiple entities in host computers connected to the system.

2.7 Communication Services

Network communication services will provide for the timely and efficient transfer of data between the public components of the individual host computers required to create virtual, interactive environment repesentations. Both local area and wide area services will be provided though routers and gateways which service multi-peer/multi-cast distribution of data.
In addition to providing time critical transfer of data required for creating a real time simulation environment, the communication services will have the provision to transfer tactical data and voice using either actual military communications or a segment of the simulation system communication services as appropriate.
Likewise, the communication services will provide for data transfer of non time critical data supporting system management and administration.

3. INTEROPERABILITY AND PROTOCOL DATA UNITS

A means has been found to assure interoperabily between dissimilar simulations in developing and defining a communications protocol. There must be an agreed-upon set of messages that communicate between host computers, the states of simulated and real entities and their interactions. This information is communicated in the form of Protocol Data Units (PDU).

The types of DIS PDUs currently required in the "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation" standard (Ref. 2) includes:

- Entity State PDU;
- Collision PDU;
- Emission PDU;
- Transmitter PDU;
- Signal PDU;
- Receiver PDU.

This list is not complete: more PDUs are defined in the draft standard (e.g. dealing with Simulation Management, Logistics) and more PDUs will be defined in the near future.
Also existing PDUs will possibly be adapted as a result of experiences and better insight.
The Transmitter, Signal and Receiver PDUs deal with the simulation of radio communications. This Radio Communication Protocol supports the simulation of both audio and data transmission by radio.
The Emission PDU shall be used to communicate active electro-magnetic and acoustic emissions, including active countermissions.

The one PDU that is always required in any simulation is the Entity State PDU (ES PDU). It contains information about the location of the entity in the simulated world and its orientatation, the issuing entity's identification and type, the time at which the data in the in the PDU is valid, as well as information required for representation of the entity's (dynamic) visual appearance (including articulated parts). Figure 2 gives an overview of the parameters of the ES PDU.

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS	
8	PROTOCOL VERSION	8 - bit unsigned integer
8	EXERCISE IDENTIFIER	8 - bit unsigned integer
8	PDU TYPE	8 - bit enumeration
8	PADDING	8 - bit unused

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS (CONTINUED)	
48	ENTITY ID	Site - 16 - bit unsigned integer
		Host - 16 - bit unsigned integer
		Entity - 16 - bit unsigned integer
8	PADDING	8 - bit unused
8	FORCE ID	8 - bit unsigned integer
64	ENTITY TYPE	Entity Kind - 8 - bit enumeration
		Domain - 8 - bit enumeration
		Country - 16 - bit enumeration
		Category - 8 - bit enumeration
		Subcategory - 8 - bit enumeration
		Specific - 8 - bit enumeration
		Extra - 8 - bit enumeration
64	ALTERNATIVE ENTITY TYPE (GUISE)	Entity Kind - 8 - bit enumeration
		Domain - 8 - bit enumeration
		Country - 16 - bit enumeration
		Category - 8 - bit enumeration
		Subcategory - 8 - bit enumeration
		Specific - 8 - bit enumeration
		Extra - 8 - bit enumeration
32	TIME STAMP	32 - bit unsigned integer
192	ENTITY LOCATION	X - Component - 64 - bit floating point
		Y - Component - 64 - bit floating point
		Z - Component - 64 - bit floating point
96	ENTITY LINEAR VELOCITY	X - Component - 32 - bit floating point
		Y - Component - 32 - bit floating point
		Z - Component - 32 - bit floating point
96	ENTITY ORIENTATION	Psi - 32 - bit BAM
		Theta - 32 - bit BAM
		Phi - 32 - bit BAM
320	DEAD RECKONING PARAMETERS	Dead Reckoning Algorithm - 8 - bit enumeration
		Other Parameters - 120 - bit unused
		Linear Accel - 3/32 - bit floating point
		Angular Velocity - 3/32 - bit integer
32	ENTITY APPEARANCE	32 - bit unsigned integer
96	ENTITY MARKING	Character set - 8 - bit enumeration
		11 8 - bit unsigned integer
24	PADDING	Unused
8	#ARTICULATIONS PARAMETERS	8 - bit unsigned integer
VARIES	ARTICULATIONS PARAMETERS	Change - 16 - bit unsigned integer
		ID - attached to - 16 - bit unsigned integer
		Parameter type - 32 - bits
		Parameter value - 64 - bits

Figure 2: *Entity State PDU*

Most important with respect to network bandwidth and delays are the Dead Reckoning (DR) parameters to enable the receiving entities to maintain a model of the position and orientation of entities that are of interest (within sight or range of the sensors of the receiving entities).

One of the engineering issues with networked simulation is the question of what DR and smoothing algorithms are to be used in relation with delay, network bandwidth, the number of entities on the network and of cause the allowable offsets for position and orientation.

4. LATENCY AND ENTITY STATE UPDATE

An electrical signal transmitted from one point to another has always a propagation or transport delay even with unlimited channel bandwidth. If this signal is transmitted via a geostationary satellite the round trip is approximately 240 milliseconds. An aircraft flying at 500 knots can travel ca. 62 m in that amount of time. If uncompensated, this amount of delay could have a negative effect on training. Channel bandwidth is another physical limitation that can add to delays and will impose limitations on the number of entities in a networked exercise, mainly due to the frequency of entity state updates.

Schaffer and Waters (BBN), and Harvey (BMH) determined average ES PDU transmission frequencies for a F-16 aircraft, performing the 5 minute Paris airshow flight demonstration (Refs. 3 and 4). They showed that with 2nd order position, 1st order orientation DR algorithms position errors < 3 m and orientation errors < 3 degrees or position errors < 1m and orientation errors < 10 degrees could be obtained with ca. 1 Hz average update rate (See Figure 3).

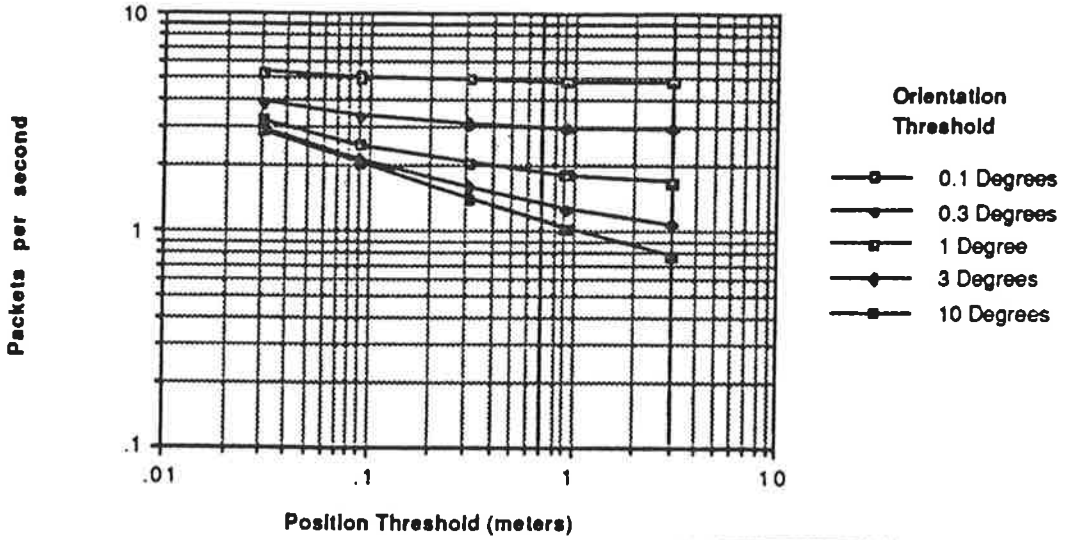


Figure 3: *Average ES PDU Transmission Frequency for F-16 Paris Airshow using 2nd Order Position / 1st Order Orientation DR Algorithm*

Goel and Morris (Northrop) performed experiments using higher order DR algorithms which incorporated time compensation techniques and local smoothing algorithms to account for varying network transmission delays (Ref. 5). Their experiments confirmed that it is possible to simultaneously maintain vehicle position and orientation "ground truth" for the most dynamic vehicles, with transmission delays up to 750 msec.

The use of dead reckoning is imperative in DIS since it decreases the number of ES PDUs to be communicated over the network. However it doesn't solve all problems caused by latency. In the field of electro-magnetic interactions there is the need to be able to simulate the "last second" effects of interactions. For that (and other) purposes a "hand-over" PDU is proposed, whereby control over an entity is handed to another host.

So far little has been said about the underlying communication services that are needed to transfer the Protocol Data Units.
Whereas the communication protocol for the PDUs describes and specifies the services of layers 6 and 7 of the ISO Reference Model, there are services required within the Transport, Network, Link and Physical layers to transmit the PDUs.
These services have been categorized as (Ref. 6):

- Reliable
- Best effort
- Multicast
- Unicast
- Real-time

With real-time is meant communication latency less than 300 msec (Figure 4).

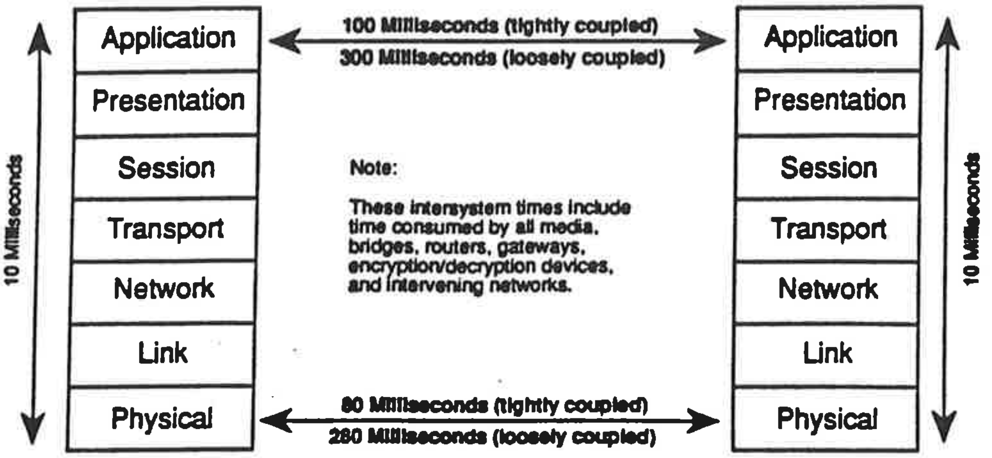
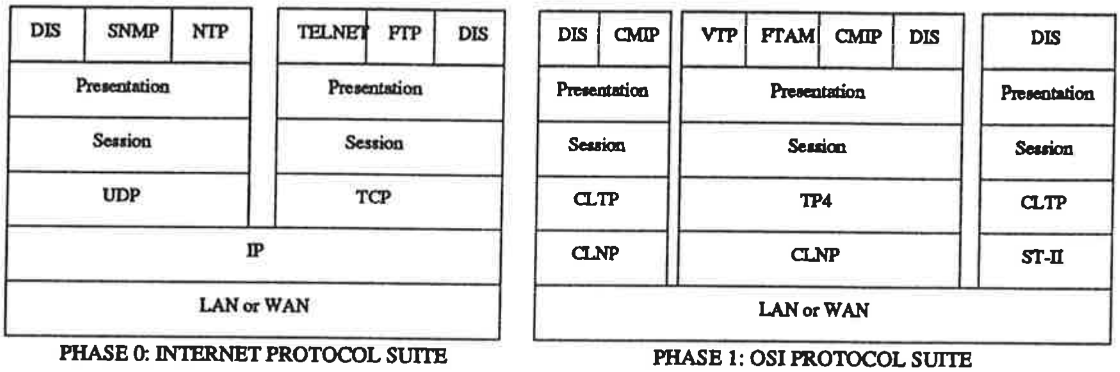


Figure 4: *ISO Reference Model and standard latency values*

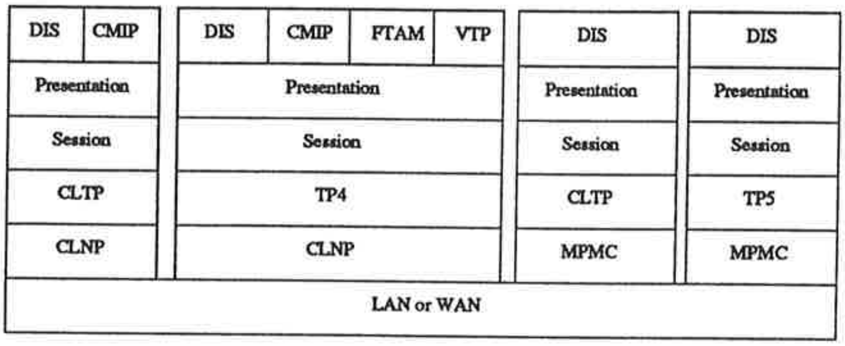
The figures 100 and 300 msec come from experience in the flight simulator industry. It is stated that 100 msec is effectively a floor latency value, because humans cannot distinguish differences in time that are less than 100 msec. That is, with a human in the control loop, there is no benefit to be gained from latencies less than 100 msec.
In situations where latency reaches 300 msec, pilots start compensating for the lag in responce. The phenomenon is known as Pilot Induced Oscillation.

The service requirements listed above and combinations thereof cannot fully be met at this moment with commercially available communication protocol suites. For LANs (Ethernet, Token ring, FDDI) only there are practical solutions today, but for WANs the combination Multicast/Real-time is not available. However, in concert with emerging technologies such as Asynchronous Transfer Mode (ATM), Synchronous Optical NETwork (SONET), Frame Relay, no doubt within some years there will be communication stacks available, which do comply with DIS, as well as with OSI standards.
Therefore a phased migration path for the DIS communication architecture is specified in Ref. 6. See Fig. 5.



PHASE 0: INTERNET PROTOCOL SUITE

PHASE 1: OSI PROTOCOL SUITE



PHASE 2: GOSIP PROCTOCOL SUITE

Figure 5: *DIS Three Phase Protocol Suite*

5. CONCLUSIONS

Although not addressed in this paper several potential benefits can be derived from networking simulators. The feasibility has been demonstrated. Some of the critical engineering issues have been addressed, but there are many more that need further research and experimentation.

6. REFERENCES

1. *Distributed Interactive Simulation Operational Concept (2.1)*, UCF Institute for Simulation & Training (IST-TR-92-32)
2. *Military Standard (Version 2.0): Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation*, UCF Institute for Simulation & Training (IST-CR-92-12)
3. Schaffer, R. and R. Waters 1991, Dead Reckoning Algorithms and the Simulation of High Performance Aircraft, BBN Systems and Technologies, White Paper ASD 91-015
4. Harvey, E. and R. Schaffer 1991, The Capability of the Distributed Interactive Simulation Networking Standard To Support High Fidelity Aircraft Simulation, *Proceedings Eleventh Interservice/Industry Training Systems Conference*; Orlando, Florida
5. Morris, K. and S. Goel 1991, Interim Report for Long Haul Data Transmission System, Northrop Report NOR 91-28, Northrop Corporation
6. *Military Standard (Draft): Communication Architecture for Distributed Interactive Simulation (CADIS)*, UCF Institute for Simulation & Training (IST-CR-92-6)

ESA-WPP-039

PROCEEDINGS

of the

Second ESA Workshop on Simulators for European Space Programmes

25-26-27 November 1992

ESTEC, Noordwijk, The Netherlands

Organised by:

EUROPEAN SPACE AGENCY Simulation & Electrical Facilities Division - Technical Directorate