

IETF Large-Scale Multicast Applications (LSMA) Working Internet Draft  
<draft-myjak-lsma-scenarios- 26 March 199702.txt>

Steve Seidensticker, ed., seiden@netcom.com  
W. Garth Smith, wgsmith@metavr.com  
Michael Myjak myjak@ist.ucf.edu

# Scenarios and Appropriate Protocols for Distributed Interactive Simulation

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as ``work in progress."

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt" listing contained in the Internet- Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited.  
Current schedule:

June 1996. Meet in Montreal, approve charter, refine goal documents and finalize the appoint of editors for documents.

Sept. 1996. Produce initial draft.

Dec. 1996. Review document drafts and accept changes and comments on the documents.

March 1997. Publish documents as Informational RFCs. Close group.

Mailing lists for comments:  
ietf-dis@bacon.gmu.edu  
ietf-dis-request@bacon.gmu.edu

## Abstract

We describe distributed interactive simulation (DIS) scenarios from the vantage point of hardware and software vendors who would need to address the network implications and requirements to enable large scale networked multiplayer virtual worlds. This document is meant to migrate the knowledge of the traditional Department of Defense (DoD) modeling and simulation community into tangible design metrics for the commercial networking community [2]. [Note: The term "DIS" is used here generically to describe

the type of simulations used to implement these scenarios. It does not imply the use of IEEE 1278.1 protocols[1], frequently referred to as DIS protocols.]

## **1. Introduction**

Distributed simulations are collections of individual simulations linked at a peer level. That is, there is no client/server or centralized kernel. Each simulation application maintains its own copy of a common virtual environment. Representations of this environment (e.g. terrain databases) are distributed by various means to all simulation applications prior to any real time operation. Issues associated with the distribution of the static environment are not within the scope of this discussion.

During real time operation each simulation application models the behavior of the simulated entity(s) for which it is responsible. Whenever specific events occur in the virtual environment or when the state of the simulated entities have changed significantly, the simulation application transmits this new information to the other simulation applications on the network. This information is encapsulated in packets and is transmitted via the UDP/IP datagram service. Anomalies and lack of precise causality due to dropped packets are tolerated in order to get the latency needed for real time operations.

Effective distributed simulation depends on low latency between the time a new state/event occurs for a simulated entity to the time that that state/event is perceived by another entity that must react to it (e.g. two high performance aircraft maneuvering to get in position to launch weapons at each other). In contrast, during event driven simulations increased latency merely slows the simulation down but causality is maintained. In real time simulations excessive latency destroys interactions and the simulation collapses.

The recommended practice for communications architecture (IEEE 1278.2) indicates that the underlying communications structure should provide 100 ms or less latency for packet exchange for closely coupled interactions between simulated entities (e.g. high performance aircraft in a dog fight). This requirement is based on human reaction times that have been the basis of human-in-the-loop (HITL) flight simulator designs for many years. This requirement eases to 300 ms for loosely coupled interactions (e.g. simulated voice radio).

To some extent real time simulations can compensate for excessive latency by extrapolating continuously changing values (e.g. dead reckoning of entity movement based on initial position plus velocity and acceleration). However, discrete events (e.g. voice exchanges between humans on a simulated radio net) cannot be predicted and therefore no latency compensation is possible.

### **1.1 Types of Data Exchanged**

In real time distributed simulations any data may be sent between applications, however the following categories of data dominate and tend to tax network services:

Entity State -- Information includes appearance, location, velocity, orientation, accelerations, and position/movement of articulated parts of

simulated entities. Location and movement are dead reckoned and this packet is only sent to correct dead reckoned parameters. It may also be sent periodically as a heartbeat, or to compensate for lost packets. Radically maneuvering entities transmit up to 15 packets/second. An average rate is two/sec for land vehicles and five/sec for aircraft.

Emissions -- Information includes point of origin, power, frequency, direction, scan pattern, and other parameters of electronic or acoustic emissions. This information is used to stimulate simulated sensors capable of detecting and responding to such information. In electronic warfare (EW) environments these parameters change frequently. In an active EW environment emission packets are sent as frequently as entity state packets.

Bit Stream Packets -- These represent voice samples, computer-to-computer communications, images, or any other digital bit stream. These are heavily used in simulations that include voice radio, intelligence, and tactical command and control systems.

Environment -- Atmospheric or oceanographic data is broken up into a series of packets to describe changes in the simulated environment. The update rate is relatively low, but the number of packets needed to represent complex patterns induces a significant network load.

Fire & Detonation -- Packet pairs carry the information needed to describe the firing of a ballistic (unguided) weapon and the detonation of the projectile. The amount of information per packet is modest and the total packets are limited to the amount of ammunition the participants can carry, but weapon firing tends to come in bursts and those bursts can impact network performance.

In real-time simulations all the data listed above shares an important characteristic that is often overlooked when designing reliability mechanisms. The data is perishable. It becomes stale quickly. Most reliability mechanisms attempt to retransmit the original data to correct for packet loss. This approach may be required for convention applications (e.g. file transfer), but in distributed real-time simulation such recovery is of little use. A better approach is a recovery mechanism that retransmits a fresh version of the data in a lost packet.

## **1.2 Network Requirements**

The following are extracted and summarized from a companion informational RFC by Pullen, Bouwens, and Myjak [4].

### **1.2.1 Multicasting**

In early DIS simulations all applications simply broadcast packets to all other applications in the exercise using the IP broadcast address in UDP datagrams. More recent simulations use static multicast addresses to segregate different kinds of packets (e.g. entity state, emissions, bit streams). This reduces the processing required to throw away packets by those applications that cannot process them.

While static multicasting is a significant improvement over broadcasting, it is not adequate for the large scale simulations. For that a mechanism

must be found that can efficiently route packets only when and where they are needed. Dynamic multicasting is the key. But to use dynamic multicasting it must be supported by the communications community.

Much has been said and written about dynamic multicasting. There is no single view of it. Dynamic multicasting can have many characteristics. Different applications will emphasize different characteristics. Those most important to the DIS community are:

- \* Ability to support and manage thousands of simultaneous multicast groups.

- \* Sustain join/leave times of less than one second at rates of hundreds of join/leaves per second with complete end-to-end propagation.

This must be done using a many-to-many paradigm in which 90% or more of the group members act as senders and receivers within any given multicast group.

### **1.2.2 Real-time Packet Delivery**

Packets must be delivered with low packet loss and predictable latency on the order of a few hundred milliseconds, after buffering to account for jitter (variation of latency) such that less than 2% of packets fail to arrive within the specified latency, in a shared network.

### **1.2.3 Resource Reservation**

Simulation exercises using the Internet Protocol Suite (IPS) will need to reserve a maximum overall networking capacity that would be dynamically shared among various groups of information flows. With regards to distributed simulation environments, this implies that information flows will need to be dynamically grouped in relation to multicast groups, regions of interest, or some possibly some other paradigm as the exercise evolves.

### **1.2.4 Resource Sensitive Routing**

Forwarding datagrams from one network node to the next is performed by routing protocols such as Multicast Open Shortest Path First (MOSPF). In order to make the overall network function, many knowledgeable IETF people believe a routing protocol is needed that determines paths through the network within the context of a quality of service requirement.

## **2. Scenarios**

### **2.1. Local, small scale, real-time simulation.**

A dedicated 10 megabit/second Ethernet 10Base-T Local Area Network (LAN) has been established for a 200 entity exercise. The simulation platforms are Silicon Graphics (SGI) Indigo 2 Extremes running IRIX 5.3, SGI Maximum Impacts, and SUN Solaris 5.4 SPARC Ultra UNIX workstations. Three stand-alone hosts running an embedded real-time operating system are functioning as high speed aircraft simulators. The aircraft simulators provide high resolution out-the-window views of the virtual world. All simulations are using the IEEE 1278.1 protocol for entity to entity interaction. Primary entity information is being passed as UDP/IP datagrams via the UNIX UDP port 3000. All simulation hosts are located on the same

subnetwork on the LAN. All simulations are using a 20 X 20 kilometer terrain database so as to orient in a common virtual world.

All UNIX workstations are configured with Internet Group Management Protocol (IGMP) and all simulations are using static IP multicast. No reliability via retransmissions is provided at the network transport protocol level as no standard for reliable multicast has yet been adopted by the community [3]. A small number of predefined multicast groups are being used to delineate network traffic to interested users. These groups do not change during the course of the exercise. As in typical DIS scenarios, all simulation applications are acting like senders and receivers. Some simulation applications emit multiple entities (up to 40) while others are responsible for only a single entity's behavior.

The majority of entities are low speed ground vehicles transmitting IEEE 1278.1 Entity State packets of 1152 bits at an average of two packets per second. These ground entities are controlled by a rule based system that mimics behavior of transportation on four of the SGI simulation hosts. One of the rule-based simulation applications is generating six high speed rotary wing aircraft generating on average six Entity State packets per second. Occasionally, these rotary wing entities emit bursts of 704 bit Fire packets with subsequent emission of 832 bit Detonation packets (on the order of 20 to 100 packets a second for both Detonation and Fire). All Entity state based information must arrive at all the designated receivers at the 100 millisecond constraint to remain real-time.

The three high speed aircraft simulation emit from 5 to 10 Entity State packets per second. Additionally, the aircraft are using experimental radio communication bit stream packets for transmitting voice between the users. Pregenerated background voice traffic is being transmitted along with the pilot communications. An air-borne control aircraft simulation transmitting voice to the pilots is also issuing voice bit stream packets. These packets are on the order of several hundred bytes and are randomly dispersed through the exercise with occasional bursting. To limit the impact of the voice traffic on simulation applications that cannot process such traffic, a predefined static multicast group is used for all voice communications.

One of the simulation applications provides weather changes to exercise participants via a predefined static multicast group. Due to the very large size of the precipitation and wind data (five megabits), the transmitting applications fragment the data into manageable packet sizes (i.e., up to the Ethernet datalink size limitation). Receiving applications reassemble the packets into the formats needed by the simulation. The weather data need only arrive at the specified users host at 10 minute intervals to remain real-time.

The scenario is periodically compromised when the network traffic exceeds the 10 megabit/second rate. This usually occurs when the radio and weather packets and detonations are all simultaneously occurring. Other loss of network information occurs when the host receiver buffers get overloaded and drop packets when receiving bursts of voice and weather data simultaneously. Exercises that experience such loss are considered invalid and are replayed.

## **2.2. Synthetic Theater of War (STOW)**

The Defense Advanced Research Projects Agency (DARPA) is funding the expansion of the simulation in scenario one above to a much larger scale. The original goal of the STOW program was to demonstrate in real time the simulated interaction of 100,000 or more entities via applications at 50 or more sites. The general purpose is to rehearse theater level military operations using actual players. Budget and other constraints have reduced the goals of the demonstration significantly. However the techniques, technology, and architecture used in the demonstrations must show that they will support simulations of larger scale.

The STOW program started with the IEEE 1278.1 message based protocol for the exchange of data but is shifting to a new approach being developed by the Defense Modeling and Simulation Office (DMSO), called the High Level Architecture (HLA). The HLA moves the interface from the messages to an API that resides between the simulation applications and a layer of standard services called the Runtime Infrastructure (RTI). The RTI itself is distributed. Parts of it reside with each application and parts of it may reside in processors dedicated to providing RTI functions.

The RTI services include the exchange of simulation data between the applications. The mechanism employed is an object request broker (ORB) that sets up communications channels between publishers and subscribers. Whenever a publishing application updates an attribute the RTI moves it to the subscribing application(s). The RTI relies on standard UDP/IP services to move the data. The HLA publish/subscribe mechanism is quite sophisticated and allows subscriptions based on the values of attributes (e.g. give me the location of all air vehicles within sector x). The RTI determines the flow of data based on the publications/subscriptions and establishes data channels via multicast addresses that carry the flow from senders to groups of receivers. As the values change on which the subscriptions are based, the membership of the multicast address based on that value range must also change. This change must occur rapidly if the data flow is to remain correct and the simulation is to remain valid.

Although the ORB used by the HLA RTI is different from the message based protocol used in older simulations, the fundamental multicast address requirements will not change. As long as the data sources, the destinations, and circumstances under which the data flow remain the same, the multicast service requirements will remain the same.

## **2.3. VR on the Internet**

Phase I -- You are interested in cathedrals and you have chosen that subject for the paper you have to present next week in your European history class. You jump into the WWW with your favorite browser and ask Lycos to find you something about cathedrals. In a few seconds it comes back with a list of several hundred articles. You click on one that promises exciting pictures. The view of the front of Chartres that fills your 20" monitor is just amazing. The spires, the arches, the detail look almost real. You zoom in on the entrance but the detail disappears into blocks of pixels.

This is the state of the WWW today. It provides you copies of two dimension representations that someone else has created and put onto the web for you to fetch.

Phase II -- You are again looking at the image of Chartres. Only this time you click on a "move forward" control. The entrance gets larger. The detail of the doors comes into view. As you approach the doors swing open and you enter. The vaulted ceiling and stained glass windows are magnificent. You look right and left. You move about the interior and examine the detail.

This is the state of the WWW tomorrow morning. That is, the capability of representing an object on the WWW in three dimensions, viewing that object from any point in space, along with simple animations (doors opening on approach) is available on a few WWW servers and browsers using the Virtual Reality Modeling Language (VRML).

Phase III -- You turn back to the altar. Now see and hear a priest saying mass. He stops and turns to the choir. You hear the strains of a familiar hymn. You look at the choir. They look quite real. The quality of their voices convinces you that they are not a professional group. One of them turns to you and beckons. You move toward them and take a spot in the front row. You begin singing into the microphone attached to your computer. Your voice is combined with theirs' and is heard by the priest, the other members of the choir and a few people you notice in the pews.

This is the state of the WWW the day after tomorrow where individuals will be able to share common virtual environments. In these shared environments individuals will be represented by what the industry is calling "avatars". Such avatars will be under the direct control of the real person they represent. For an excellent extrapolation of where this concept might go, read Neal Stephenson's "Snowcrash."

Although such shared virtual environments are not simulations in the classic sense, they have many of the same requirements of the supporting communications structure. That is, they are very similar MC requirements.

#### **2.4. Recreations of Historic Battles**

The year is 2010 or thereabout. The world's major powers have been at peace since the resolution of the Balkans conflict in 1997. Several generations have grown up without the taste of battle. But there is something in the psyche of these generations that yearns for the excitement of engaging an enemy with skill and determination, but without the shedding of their own blood. This yearning has been behind the development of more and more realistic real time simulations. It has culminated in the simulated recreation of historic battles on a grand scale.

A culture has grown up around these simulations. Above all they strive for historic accuracy. The simulated planes, ships, tanks, horses, and weapons have the same capabilities as the original. Logistics, planning, and initial conditions are the same. The participants are trained to the same point as their original counterparts. They are tested and qualified before they can participate. Some take on the names and habits of real characters. Many assemble at annual conventions to exchange tales and see the latest in simulation equipment.

These simulations have become a very popular spectator sport. Millions of viewers watch the battles unfold from any vantage point that they choose. They can go into the command centers. They can tether themselves to any

participant and see and hear what he sees and hears. They can tune in to various commentators in different languages for an explanation or critique of what is going on at various points. The recreations are both scheduled and reported on the popular net sport pages.

The scenario described below is the recreation of a series of B-17 raids that the US Eighth Air Force made over Europe in 1943. These raids were fiercely and effectively defended by the Luftwaffe and German anti-aircraft artillery (the dreaded flak).

The participants are a mix of humans and software models. The humans tend to take the more interesting roles. The software models are sophisticated enough that they cannot be easily distinguished from human participants. Total number of participants roughly corresponds to the number of participants in the original battle and varies between 100,000 and 500,000. Number of spectators ranges in the tens of millions and is governed by the factors that control the watching of any sport.

The simulation equipment ranges from simple VR goggles to elaborate cockpits and gun turrets with sophisticated motion and sound cuing. All simulations share visual cues that are indistinguishable from what the unaided eye can see. The participants own or rent the hardware and software. As with any avocation the cost varies depending on taste and finances. Participation is not governed by cost but by the dedication and capability of individuals.

The participants and spectators are located throughout the world and are connected on a virtual net that reaches into each participant's and spectator's home. Crew station simulations that are part of one simulated platform (e.g. a single B-17) are themselves distributed.

Sophisticated multicasting schemes, clever interest management algorithms, and simulation tricks have effectively removed any limits to the potential scale of distributed simulations. Local bandwidth limitations sometimes restrict what a spectator can see. Dropped packets and latency spikes occasionally cause anomalies, but they are seldom serious enough to effect the outcome of the simulation.

## **References:**

[1] IEEE Standard for Information Technology - Protocols for distributed interactive simulation Applications, IEEE Std. 1278-1993.

[2] <http://www.herring.com/mag/issue30/games.html>

[3] <http://www.tascnets.com/mist/doc/mcpCompare.html>

[4] Pullen, J., M. Myjak and C. Bouwens, "Limitations of The Internet Protocol Suite for Distributed Simulation in the Large Multicast Environment", IETF Large Scale Multicast Applications Working Group, draft-ietf-lsma-scenarios-xx.txt, work in progress.