A NETWORK ARCHITECTURE FOR LARGE SCALE VIRTUAL ENVIRONMENTS

*Major Michael R. Macedonia, U.S. Army
Computer Science Department
U.S. Naval Postgraduate School
Monterey, California 93943-5118
macedonia@cs.nps.navy.mil

David R. Pratt
Computer Science Department
U.S. Naval Postgraduate School
Monterey, California 93943-5118
pratt@cs.nps.navy.mil

Michael J. Zyda
Computer Science Department
U.S. Naval Postgraduate School
Monterey, California 93943-5118
zyda@cs.nps.navy.mil

1. Introduction

The purpose of the paper is to outline the problems and a proposed solution to the design and construction of large scale distributed simulations. In particular this paper addresses the networking software architecture for large scale virtual environments (VEs). We suggest a method that exploits the temporal, functional and spatial relationships of real-world entities for partitioning VEs by associating network multicast groups with entity areas of interest.

The motivation for our effort is to expand the capability of virtual environments to serve large numbers (more than 1,000) of simultaneous users. Interest by the government, military, and telecommunications industry in large distributed virtual environments has been rapidly growing. For example, the US Army has plans for the Louisiana Maneuvers (LAM) initiative later in this decade which envisions 10,000 to 100,000 autonomous and human players participating in a simulation over a global wide-area network [13].

Advances in computer architectures and graphics, as well as standards such as the IEEE 1278 Distributed Interactive Simulation (DIS) and BBN SIMNET protocols have made small scale (less than 300 players) realistic man-in-the-loop simulations possible [5,8,11,]. Unfortunately, SIMNET, which was developed for small unit training, and its descendant, DIS, are currently not suitable for large scale VEs.

2. Practical Problems with the DIS Protocol

There are several major problems associated with scaling the current suite of DIS protocols for very large VEs:

Enormous bandwidth and computational requirements for large simulation. In schemes such as SIMNET and DIS, a simulation with 100,000 players would require 375 Mbit per second (Mbps) of network bandwidth to each computer participating in the simulation, an unrealistic requirement for an affordable system in this decade [7].

Lack of an efficient method of handling static objects. Large numbers of static entities such as bridges and buildings may change with respect to an event (e.g. an explosion). These and other stationary objects (e.g.“dead entities”) must send update messages at regular intervals to inform the participants of their current state.

M**odels and world databases must be replicated at each simulator.** No mechanism in DIS exists to distribute objects on demand. For large scale simulation, this is a necessity, particularly when the simulators are heterogenous, controlled by different organizations, and little coordination is expected prior to an exercise. Furthermore, it is not feasible nor efficient for each simulator to store every model and database for a 100,000 entity simulation. For example, an F15 simulator normally does not need to concern itself with naval vessels -- unless some unique scenario has the aircraft flying near or over the ocean.

3. Reasons for Problems

Event and State message paradigm. A basic requirement for DIS has been that the simulation of the VE be, as a whole, stateless - data is fully distributed among the participating hosts and entities are semi-persistent. Therefore, every entity must be made aware of every event (e.g. a missile detonation communicated by a Detonation Protocol Data Unit or DPDU) just on the chance it may need to know it. According to the protocol, an entity must, on a regular basis, communicate all of its state information (an Entity Protocol Data Unit or EPDU) to every member of the group - even though the data contained in the EPDU is often redundant and unnecessary (e.g. aircraft markings). More importantly, these “keep alive” messages can comprise 70% of the traffic for large scale simulations [14].

This paradigm as applied in DIS does not take into consideration that different simulated systems have different real-world sensing capabilities that translate into each entity’s VE data requirements. In a large VE, it is unlikely that two entities representing ground vehicles separated by 200 Km need to be aware of each other. Yet, under the current architecture they must inform each other of state changes and updates.

The rationale for this is to avoid the reliability problems of a central server, to simplify communication protocols, and minimize latency while guaranteeing that hosts entering a simulation would eventually build their entity database through entity state and event messages. Furthermore, the use of broadcast EPDU updates is part of the effort to maintain consistent view among the simulators within a particular tolerance.

Real-time system trade-off's. Reliability must be compromised for real-time performance in large distributed groups. This is because in order to be truly reliable the system requires the use of acknowledgment schemes which defeats the notion of real-time, particularly if a player host must establish a virtual connection with every other entity host to ensure that each received data correctly. The corollary is that a real-time environment should avoid transactions between entities since this requires reliable communications. Furthermore, schemes that use a central database do not work well in a large VE due to I/O contention. For example, AT&T's Imagination network limits the number of concurrent players in a game to four because they are centrally served and bandwidth is limited to the speed of modems (less than 28 Kbps).

No "middleware" layer. There does not exist a protocol component that mediates between the distributed VE applications and the network. The current DIS paradigm implies the use of a bridged network because every message is broadcast to every entity. However, internetworking (routing over the network layer) is necessary for large scale simulations because it provides the capability to use commercial services as opposed to private networks to bring together diverse, geographically dispersed sites, use different local network topologies and technologies (e.g. Ethernet and FDDI), and take advantage of "rich" topologies for partitioning bandwidth, providing robustness and optimization of routes for minimizing latency. Confining DIS to the data link layer requires the use of bridges which are a magnitude slower to reconfigure after a topological change than routers while the number of stations are limited to the tens of thousands. A network with routers is limited to the numbers accommodated by the address space [10].

Origins as small unit training systems for Local Area Networks (LANs). Many of these problems devolve from the fact that until recently DIS and SIMNET were used exclusively for small scale training simulations so that it was relatively easy to insure that the VE components would have homogenous sets of models and terrain databases. The lack of middleware stems from the monolithic nature of these small scale environments which could be distributed using a single LAN. These origins have also influenced the current assumptions about the density and rates of activity of entities in large scale simulations that do not necessarily match the real world. Players in SIMNET participated for short periods (several hours) and were highly active because the purpose of the simulation was to train crews in coordinated drills. Furthermore, the density of systems with respect to the simulated area of play was high because that best represented a small unit engaged in close combat and because of the difficulty in using large terrain data bases.

4. Exploiting Reality

We believe that it is incorrect to strictly extrapolate the SIMNET and DIS experience to large scale VEs. For example, it is wrong to assume that a terrain "box'" the size of Fort Knox would have 10,000 weapon systems - an order of magnitude too many. Furthermore, we can exploit aspects of the real-world such as areas of interest and advance rates to efficiently use multicast groups, eliminate EPDU keep-alive updates, enhance the reliability of large scale VEs, and reduce overall bandwidth requirements.

5. General Observations

Real world battles now have a low density of systems. The median historical density for all armies is 82 weapon systems per kilometer of front [18]. However, though small units are usually organized in linear frontages, a large unit like a division extends in depth with large gaps among units. A conservative estimate for the width and depth of a US heavy division in the defense is 40x20 Km or 800 Km^2.

82 weapons systems/Km x 40 Km = 3280 weapon systems
3280 weapons systems/800 Km^2= 4.1 weapon systems/ Km^2

Another way to calculate densities is to observe that a modern heavy US division has approximately 1500 weapon systems (excluding machine guns and small arms) for a density of 2 weapons systems/Km^2. Aircraft have even lower densities. During Desert Storm roughly 1000 aircraft were in the air over Iraq with an average density of 1 aircraft per 400 Km^2. Obviously some areas of the battlefield are much more congested than what these numbers suggest. However, these average densities imply that many parts of the battlefield, at any particular time, are largely unoccupied.

Ground systems do not advance very fast or often. Helmbold in his study on the rates of advance rates for land operation found that they are not predictable [17]. Furthermore, he determined that land combat operations stand still 90-99% of the time [17]. However, we do know that the slowest modern US advance was 100 m/day at Okinawa and that the world’s record in modern warfare was 92 Km/day for 4 days by the 24th Mechanized Infantry Division in Desert Storm [18,16]. Assuming that the division moved constantly for 16 hours per day this rate of advance translates to 5.8 Km/hour. Individual vehicles may move much faster but, they would not continue at high rates very long because they fight as part of units in which movement must be coordinated.

Weapons systems have a limited area of interest. An area of interest (AOI) is one in which you have the capability to observe, sense, or influence activities. For example, tanks have an AOI of less than 10 Km because that distance is about the extent to which they can practically observe other systems on the ground and destroy them with their main gun. Indirect fire system such as field artillery have bifurcated AOI’s. For example, a 155mm howitzer has a small local AOI relative to the visual range of its crew and their small arms for defense. Additionally, it has an AOI with respect to the impact area of its munitions.

6. DIS AOI Manager

We propose the use of a software “glue” between the DIS event and state PDU paradigm and network layers that is wedded to reality. The AOI manager (AOIM) partitions the VE into a set of workable, small scale environments to reduce computational load on hosts, minimize communications on network tail links, and localizes reliability problems. Furthermore, the AOIM exists with every simulator to distribute partitioning processing among hosts.

7. Multicast

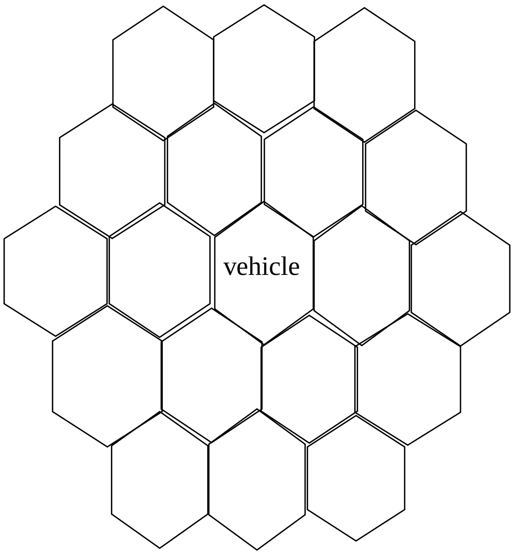
The AOIM uses spatial, temporal, and functional classes for establishing membership in multicast network groups. Multicast services allow arbitrarily sized groups to communicate on a network via a single transmission by the source [10]. Multicast provides one-to-many and many-to-many delivery services for applications such as teleconferencing and distributed simulation in which there is a need to communicate with several other hosts simultaneously. For example, a multicast teleconference allows a host to send voice and video simultaneously to a set of (but not necessarily all) locations. With broadcast, data is sent to all hosts while unicast or point-to-point routes communication only between two hosts.

The Internet Protocol (IP) Multicast protocol provides an addressing scheme that permits unreliable, connectionless, multicast service that is routable over the Internet [2,20]. From the perspective of the AOIM, IP Multicast allows the creation of transient multicast groups that can be associated with an entity's AOI. In this context, IP Multicast addresses can essentially be used as labels instead of physical destinations.

As mentioned before, this partitioning is necessary to reduce the enormous computational requirements of large scale (100,000 player) simulations. For a 1000 object exercise conducted in 1990 with SIMNET, the limiting factor was not network bandwidth, with loads running at 50%, but the local host processor performance [1]. Network simulations done by Van Hook have shown that a 90% reduction in traffic to a particular node is achievable for a 10,000 player exercise using multicast services [14].

8. Associations

For this paper we present how the AOIM would associate ground vehicle spatial AOIs with multicast addresses. We suggest partitioning the battlefield area with appropriately sized hexagonal cells. Each cell is associated with a multicast group. In the figure above we associate a vehicle with 19 hexagons that represent its AOI. Hence, it is also a member of 19 network multicast groups. The entity's host listens to all 19 groups but, with two exceptions, it sends PDUs only to the one associated with the cell in which it is located.



There are several reasons we suggest hexagons. First, they are regular, have a uniform orientation, and have uniform adjacency [19]. As the vehicle moves through the VE, it uniformly adds and deletes the same number of cells/multicast groups (5 for the vehicle above).

Secondly, a vehicle's AOI is typically defined by a radius - much like signal of transmitter in a cellular telephone system. If squares were used, we would either need to include more area than was necessary (and thus include more entities in our AOI) or use smaller grids - requiring more multicast groups - and compute which grids the vehicle should be associated with. Using hexagons with a 2.5 Km radius, the AOI above ranges from 12.5 to 8.6 Km and the area is 411 Km^2 - the approximate density of air vehicles. If the average density of vehicles was 2 per Km^2, then the entity host communicates with approximately 800 other weapon system entities. As mentioned previously, the AOI varies with respect to the capabilities of the system.

Finally, this may be convenient scheme for many large scale constructive models, such as the Army's Contingency Force Analysis Wargame and the McClintic Theater Model (MTM), that already use hexagonal cells for partitioning the battlefield.

9. Group Changes

As noted before, entities can belong to several groups at a time to avoid boundary or temporal aliasing. There will likely be few group transitions by a ground-based entity within an hour because, on average, groups of vehicles will move slowly relative to the entire VE. For the example above, if the vehicle was moving at the Desert Storm record advance rate it would transition on average a cell once an hour. The vehicle must join and leave 5 multicast groups which are associated with cells at the periphery of its AOI where change is less critical - ameliorating the effects of latency caused by joining and leaving new groups.

Moreover, we use group changes as an opportunity for database updates - similar to a paged memory scheme - in order to eliminate regular EPDU updates. We do this in a logical, distributed manner using knowledge about the age of entities with respect to their particular group.

An entity joins a group as a passive or active member. Active members send as well as receive PDUs within the group, are located in the cell associated with the group, and can become the group leader. Passive members normally do not send PDUs to the group except when they join or leave. They are associated with the group because the cell is within their AOI yet, they are not located within the cell. When an entity joins a new group it notes the time it entered and issues a *Join* PDU to the cell group with a flag indicating whether it is active or passive. The group leader replies with a PDU containing the aggregate set of active entity PDUs, the *All* PDU (APDU). A passive entity becomes an active member of a group by reissuing the Join PDU with a flag set to active when entering a cell. Departures from the group are announced with a *Leave* PDU.

The APDU may be sent reliably to the issuer of the Join PDU via a unicast protocol. With a large member distributed simulation, reliability, as provided in the Transmission Control Protocol (TCP), would normally penalize real-time performance merely by having to maintain timers for each host's acknowledgment. Moreover, flow control is also not appropriate for DIS which can recover from a lost state message more gracefully than from late arrivals. Fortunately, within the context of DIS, a certain amount of unreliability is tolerable and is mediated through the use of the dead-reckoning and smoothing algorithms [4,7]. Other applications such as packet voice and video can use adaptive techniques to handle lost packets and delays [9]. However, we can reliably send the APDU because the entity will normally be joining a group that is at the periphery of its AOI where latency is not as critical.

The group leader is the oldest member of the group. We make use of the PDU timestamps to determine the oldest member. The first active member of a group will issue several Join PDUs before concluding that its is the sole member of the group and therefore the oldest. When a passive entity determines that there is no leader it does nothing but listen for active members. A new active member of an established group issues a Join PDU, receives the APDU, notes the timestamps of the members, and keeps track of those who enter and leave.

10. Advantages

Reduced latency for new entrant learning. Assuming an even distribution of entities in our example, for each cell joined an entity must receive data for about 40 weapon systems - approximately 40 Kbits. At 10 Mbps data transfer rates, it would take 4 ms to update a new entrant versus 5 seconds under the current DIS scheme.

Reduced bandwidth requirements. This architecture eliminates the need for entity keep-alives. New entrants are informed by the Join procedure of who exists in their particular groups. Multicast association further reduces the traffic demands on the tail links by confining the scope of an entity's communication to its area of interest.

No need for a centralized server. Using the oldest member of a group to serve Join requests is logical because it is the entity that should know all of the other entities and the past events that have occurred in the group. We expect that serving the group will be relatively undemanding with respect to Input/Output processing for the group leader because of the small number of active members in a group/cell and slow transitions due to the expected real world transition rates for ground vehicles. With 40 vehicles entering and leaving a group per hour, the leader would issue, on average, one APDU per 90 seconds.

Solves the static and dead entity problem. Likely candidates for the group leader will be static entities such as those representing buildings or bridges which can change state (i.e., collapse). They will be the originating members of a spatially associated group and remain with the group for its entire existence. Moreover, static or dead entities are no longer a major burden to the VE with respect to wasting bandwidth with update EPDUs. They need only to transmit PDUs upon initialization and when changing state.

Localization of reliability problems. Large scale VEs will naturally have some degree of reliability. Partitioning the VE into groups prevents problems from impacting on the entire simulation. Currently, an entire DIS simulation involving hundreds of entities can fail because of a single rogue application because all communication is broadcast.

Maintains the current DIS semantics. The AOIM can be run as a separate thread or process and eliminates the need to change current DIS PDU semantics. The application simulating an entity is not required to have knowledge of the partitioning or the AOIM.

Not constrained to ground vehicles. We believe our framework will work equally well for aircraft, space-borne systems, and command and control systems. For example, radio channels can be associated by the AOIM with a multicast group and a voice application that simulates a particular system - relieving the simulation itself of handling real-time voice communications.

11. Conclusion

This paper describes a concept that provides a network software architecture for solving the problem of scaling very large distributed simulations. The fundamental idea behind our approach is to logically partition virtual environments by associating spatial, temporal, and functionally related entity classes with network multicast groups. This is accomplished by exploiting the actual characteristics of the real-world large scale environments that are to be simulated, and by focusing an entity's processing and network resources to its area of interest via an AOIM. Finally, we present an example of how we would implement this concept for ground vehicles. We have begun design and construction of the AOIM for use with the NPSNET 3D vehicle simulator. NPSNET is currently the only DIS compliant simulator using IP Multicast communications and is suitable for peroration over the Internet.

1 Chung, James W., *An Assessment and Forecast of Commercial Enabling Technologies for Advanced Distributed Simulation*, Institute for Defense Analysis, Arlington, VA (Oct 1992).

2 Deering, Stephen. *Host Extensions for IP Multicasting*. RFC 1112. (Aug 1989).

3 Doris, Ken. *Issues Related to Multicast Groups*. The Eighth Workshop on Standards for the Interoperability of Defense Simulations. (Mar 1993). pp. 269-302.

4 Harvey, Edward P., Schaffer, Richard L., *The Capability of the Distributed Interactive Simulation Networking Standard to Support High Fidelity Aircraft Simulation*, (BMH Associates, Inc. and BBN Systems and Technologies, Norfolk VA, Cambridge, MA. Jul 1992).

5 Institute of Electrical and Electronics Engineers, International Standard, ANSI/IEEE Std 1278-1993, *Standard for Information Technology, Protocols for Distributed Interactive Simulation* (Mar 1993).

6 Institute for Simulation and Training, IST-TR-93-20, *Communication Architecture for Distributed Interactive Simulation (CADIS) [Final Draft]*. (University of Central Florida, Orlando, FL. June 1993).

7 Loral Systems Company, *Strawman Distributed Interactive Simulation Architecture Description Document Volume 1*. (Advanced Distributed Simulation Technology Program Office, Orlando, FL, Mar 1992).

8 Miller, Duncan C.,Pope, Arthur C.; and Waters, Roland M. *Long-Haul Networking of Simulators*. Proceedings: Tenth Interservice/Industry Training Systems Conference (Dec 1989). p. 2.

9 Partridge, Craig.*Gigabit Networking* (Addison-Wesley. Reading, MA. 1994). pp. 191-193

10 Perlman, Radia. *Interconnections: Bridges and Routers*.(Addison-Wesley. NY.1992). p. 258.

11 Pope, Arthur, BBN Report No. 7102, *The SIMNET Network and Protocols*. (BBN Systems and Technologies, Cambridge, MA. Jul 1989).

12 Pratt, David R., *A Software Architecture for the Construction and Management of Real Time Virtual Environments* (Naval Postgraduate School, Monterey, CA. Jun 1993).

13 *United States Army Posture Statement*. US Army. (Mar 1993). pp. 65.

14 Van Hook, Daniel J. *Simulation Tool for Developing and Evaluating Networks and Algorithms in Support of STOW 94*. Presented for Scalability Peer Review. (Aug 1993).

15 Zyda, Michael J., Pratt, David R. et al., *The Software Required for the Computer Generation of Virtual Environments*. in *Presence*, **2**, 2. (Spring 1993). pp. 130-140.

16 Dunnigan, James and Macedonia, Raymond M. *Getting It Right*.(Morrow. NY, NY. 1993).p. 211.

17 Helmbold, Robert L. *Rates of Advance in Historical Land Combat Operations*. CAA-RP-90-1. US Army Concepts Analysis Agency. Bethesda, MD. (Jun 1993).pp. 5-2,3.

18 McQuie, Robert. *Historical Characteristics of Combat for Wargames*. CAA-RP-87-2. US Army Concepts Analysis Agency. Bethesda, MD. (Jul 1988). p. 13.

19 Samet, Hanan. *The Design and Analysis of Spatial Data Structures*.(Addison-Wesley. Reading, MA. 1989). pp. 20-21.

20 Macedonia, Michael R. and Brutzman, Donald P. *MBone Provides Audio and Video Across the Internet* in *IEEE Computer*. (Apr 1994).