1996

Networking Large-Scale Virtual Environments

Zyda, Michael J.

þÿZyda, Michael J. Networking Large-Scale Virtual Environments, Procee
þÿAnimation 96, 3- 4 June 1996, Geneva, Switzerland, IEEE Computer Soc
(an earlier version of this paper also appeared in the Proceedings of the Second International
Conference on the Military Applications of Synthetic Environments and Virtual Reality,
Stockholm, Sweden, 6-8 December 1995, pp. 119-125.)

Networking Large-Scale Virtual Environments

Michael J. Zyda
Naval Postgraduate School
Department of Computer Science, Code CS/Zk
Monterey, California 93943-5118
zyda@siggraph.org

Abstract

The virtual environment research community is focusing on the human-computer interaction model for fully-articulated 3D humans in the networked virtual environment. As that works moves forward, there is the desire to internetwork large numbers of players. The NPSNET Research Group is in the forefront of that work and is focusing on the network and software architecture necessary to support large scale VEs. This paper presents a look at our multicasting and applications layer protocol research.

Virtual Environment Research Goals of the NPSNET Project

The goal of the NPSNET Research Group at the Naval Postgraduate School is to explore the hardware and software technologies for the construction of computer generated, 3D worlds, worlds through which human players and vehicles can move and interact. The hardware technology we focus on is the computer graphics workstation, local & wide area networks, high resolution HMDs/monitors, & novel 3D input devices. We hope eventually to build a system that is "seamless", meaning that we can drive a vehicle across our terrain, stop in front of a building, get out of the vehicle, enter the building on foot, go up the stairs, enter a room and interact with items on a desktop.

Large Scale Networking

Our work in the past year has had multiple foci but the theme of this paper is networking. It is quite easy to grow out of a single workstation when constructing a virtual world, especially if you expect to have multiple players in that world. What is needed to support large-scale networked virtual environments is a standard message protocol between workstations that communicates changes occurring in the world [1]. We need an extensible, real-time-reconfigurable virtual environment protocol. The system should support late arriving players, providing those players complete updates of the current state of the world.

Small-Scale Networking - State-of-the-Art

Today we build small systems, systems where *all* players on the network have the same world models, and receive all world changes as time moves forward in the virtual environment "action". In each workstation's memory is a complete model of the state-of-the-world.

What do we mean by a small system? We mean a low numbers of networked players, on the order of 250 - 500 players. Current SIMNET and DIS-compliant systems look like this and use Ethernet & T-1 links for connectivity.

Each workstation in the small network VE has a complete model of the world and receives all packet updates for any state change. This is based upon the SIMNET model, which is carried over in DIS.

Large-Scale Networking Desired - State-of-the-Art

The focus in VE research is now moving towards systems with large numbers of players, on the order of 10,000 to 300,000 players. In such systems, we can no longer propagate world state changes to every player on the network. We need to start thinking about having the player's workstation processing only "relevant" packets. The primary capability we need to have in our systems is the "generalized" capability to publish/read/consume information packets, with those packets somehow being "intelligently addressed", i.e. only provided to workstations that are part of some sort of logical grouping.

"Generalized" means extensible, meaning we can add in new packets for information readily or that we have the capability to send any type of information across the net, without failures or problems among already connected players. It is the "generalized" information transmission that we do not do today in the currently demonstrated, networked virtual environments.

The NPSNET Networked Virtual Environment

Now above is our desires. We need to talk about a real system, NPSNET. The NPSNET system is our testbed for exploring the technological challenges involved in building large-scale, networked virtual environments. The goal of the project is to develop a virtual world shell that allows one to visit any area of the world for which a terrain database is available and to interact with any interactive or autonomous players found "in the system".

NPSNET is interoperable with the SIMNET system and DIS-compliant visual simulators. The wide area networking and internetworking of NPSNET is a large part of our efforts.

NPSNET-IV is written using the Silicon Graphics visual simulation toolkit Performer and utilizes the parallel processing capabilities of multiple processor IRIS workstations to accomplish polygon culling, rendering, PDU processing. NPSNET-IV can model and visually display vehicle (air, sea-going and ground) and human movement and interaction in the DIS environment.

Network Performance with NPSNET

NPSNET-IV has been demonstrated with wide area connectivity across the Defense Simulation Internet (DSI-Net). In addition, NPSNET-IV has also been demonstrated playing across the Internet (yes, where you get your E-mail ...).

We have done experiments on the DSI-Net. Simulation traffic peaked at 168 packets per second accounting for 2.5% of Ethernet and 16% of T-1 bandwidth. Extending these numbers out, we maybe can support about 250 to 300 interactive players using such a network.

Network Performance Scalability

We are also looking at networking technology beyond Ethernet and T-1 links - we assume Sonet for the WAN (2.4 GBit/second) and ATM for the LAN (140 MBit/second). Sonet at 2.4GBit/second supports some 1.3M messages per second, using 230 byte entity state PDUs. ATM at 140 MBit/second supports some 76K messages per second.

To evaluate these numbers, we need some additional information about DIS packets. Slow moving ground vehicles generate two messages per second. This means ATM can support some 38K ground vehicles and Sonet can support 650K ground vehicles. Fast moving air vehicles generate five messages per second. For Sonet this is 260K air vehicles. For ATM, this is 15K air vehicles.

So we can support from 15K to 38K players with currently predictable LAN architectures (ATM) and from 260K to 650K players if we assume Sonet-like technology. We have made some assumptions:

– 100% of network bandwidth is utilizable.

– Current DIS packet size still used.

– All workstations look at all packets.

VE Software Scalability Statement

A main premise of our research groups is that virtual environment software architectures can exploit wide area multicast communications and entity relationships to partition the virtual world and enable the development of scalable distributed interactive simulations. Our goals with respect to this statement are several including a scalable network software architecture for virtual environments, Scalability measured in terms of thousands of interactive users, and application to real-time simulations for military training.

Problems with DIS

The origin of our thoughts with respect to networking large-scale VEs come from our perceived problems with the DIS standard. It is our view that the current SIMNET/DIS architecture does not scale well. DIS researchers already see enormous bandwidth and computational requirements for large scale simulations that have been built on DIS. This comes from the fundamental design of DIS where models and world databases are replicated at each simulator. There are additional flaws with DIS is that it multiplexes different media at the application layer, exactly the wrong place to do so.

There are several design flaws in DIS. One of the major ones is that with the event and state message paradigm there are no persistent objects other than the starting database. This requires a lot of message traffic for "keep-alive" messages, even for items no longer in play. Another flaw is that in its use of broadcast communications DIS does not internetwork. Being able to play across the internetwork layer is essential for large-scale VEs. The flaws in DIS do not prevent useful work from being accomplished with DIS-based VEs but rather are an indication of its origin as a small unit training protocol for Local Area Networks (LANs).

Multicast

Our recent efforts have been to being looking at multicast communications as part of the solution for large-scale VEs. Multicast services allow arbitrarily sized groups to communicate on a network via a single transmission by the source. In addition, multicast allows an efficient integration of other media - voice, video and imagery. We can internetwork (route over the network layer) with multicast. So multicasting looks quite promising for use in large-scale VEs. The question then is with what software architecture?

Exploiting Reality

One of the most obvious design decisions for large-scale VEs is that we need to get away from the monolithic, broadcast world. In the real world, which virtual environments emulate, entities have a limited "areas of interest". We can exploit this with multicast communications. To do so, we need our VE interactions to be mediated by a new software layer, a layer we call the "area of interest manager (AOIM)". The AOIM partitions the world into classes or groups. Some of our proposed partitioning schemes are spatial (geographic groupings based on locality - terrain cells), temporal classes (real-time - entity state, periodic - system management, non-real-time - low-resolution entities), and functional (voice communications - radio channels, entity kind - aircraft). An area of interest manager then is a software layer between the DIS application and network that is connected to reality via the partitioning. We partition the simulation into workable chunks to reduce computational load on hosts and minimize communications on tail links.In keeping with our design philosophy, we distribute partitioning algorithms among hosts and do not rely on a central AOIM server.

Simulation

We have much more work to do to finish developing our AOIM integration into NPSNET but we have done some initial simulations. The question we were trying to answer with our simulation is, does partitioning using our architecture reduce bandwidth and computational requirements for large-scale virtual environments? A second question is, does the architecture scale and if so how well?

For the simulation, we took real world data from the National Training Center from a battle scenario. We used a constructive model on the real world data to fill in the player positions and actions. We ran a simulation of our Area of Interest Manager and kept records on the distribution of players among the hexes.

In analysis, we found the peak multicast traffic to be around 2Mbits/second, just over T-1 speed (flat for the hex sizes). We found a peak of 1,500 players in an area of interest for multicast. Our simulation showed an apparently unbounded number of players (packets) per machine for broadcast, with the application having to deal with these packets.

VE Network Protocol Research Plan

The goal of our work today in the NPSNET Research Group is to step back from DIS technology a bit and to design a virtual environment software architecture and network protocol capable of supporting thousands of networked players. In addition, we plan to produce tools that allow us to rapidly reconfigure the network applications protocol and integrate that new protocol into already running virtual environments. Our plan of research includes the following tasks.

Rapidly Reconfigurable Virtual Environment Network Protocol

The development of a generalized virtual environment network protocol is one of our primary tasks. This is a two-phase project that has a very long term goal with an enormous payback if carried out successfully. We plan to build on our previous work with DIS so that we can chart a well-grounded course.

Formal Methods for Specifying the Virtual Reality Transfer Protocol (vrtp)

The first task we plan for our work on developing what we call the vrtp or virtual reality transfer protocol is to come up with a formal method for specifying DIS in its current form. We are building a C++ code generator that reads the formal network protocol specification and generates a C++ class library that reads/writes the protocol as defined. The reads will place the read-in data into structures that were specified in the formal specification. The writes will put the data onto the network in standard form. Once we have the class library generated, the ideal next step is to recompile the library and link to NPSNET without any (or minimal) code changes in NPSNET.

Once we are successful with this stage, we then need to try another DIS-compliant system with the libraries. One system we have access to is the BBN ModSAF source code but other DIS-compliant systems are possible targets for utilization of the generated class libraries.

Once this phase succeeds, an experiment we can then perform is to change the formal protocol definition and then see if we can automatically recompile/link the modified simulators. We can even test redefining DIS smaller during this effort. We plan to build a GUI-based editor for the rapid specification of the VE network protocol, along with the code generator.

Rapidly Changing Network Protocols

The above tasks are based on current technology, with a time-consuming recompile/link step. Once we have achieved success at that level, we plan to change the protocol and the GUI-based tool such that as the formal protocol is changed, the new packet definition can be “posted” to the network while all systems are running. This means that the simulators reading the formal language don’t crash when they receive the new packet definitions and protocol definitions but rather reconfigure themselves “while running” to be able to read the new type of data.

This means that we don’t recompile to read/write the new packets but rather that the instructions for their use are encoded/decodable in the new packets sent them. This change is a major departure from the way things are currently done in DIS and hence the reason for its separation into this second phase. Similar ideas are found for this in the AT&T TeleScript protocol and we plan to exploit/modify them for our efforts.

The Area of Interest Manager Software Layer

As stated above, current DIS simulators broadcast all state change packets to all players. We plan to redesign the basic virtual environment state change propagation software and network architecture along lines similar to those described in [2]. In that paper, it is proposed that state change packets be multicast, with multicast groups being assigned to packets via a software layer called the area of interest manager (AOIM). The idea behind the AOIM is to assign packets to multicast groups based on spatial, functional or temporal criteria. The AOIM would direct packets say to geographically co-located players as opposed to all players in the entire simulation. Our task for this effort will be to modify the current NPSNET software system to include an AOIM layer and to test various AOIM group assignment schemes. We hope to be able to state appropriate group assignment criteria for a variety of military simulations. This work is of long-term impact and very important. Its payback may be immediate if implemented successfully. We do not see the current DIS community working in this direction as most of the DIS community is held to strict performance and demonstration timelines.

Acknowledgments

This work would not have been possible without the support of our research sponsors including ARPA, ONR, USARL, DMSO, US Army TRAC and USA TEC.

References

[1.]“Virtual Reality: Scientific and Technological Challenges,” Editors: Nathaniel I. Durlach and Anne S. Mavor, Committee on Virtual Reality Research and Development, National Research Council, National Academy of Sciences Press, Washington, DC 1994.

[2.]Macedonia, Michael R., Zyda, Michael J., Pratt, David R., Brutzman, Donald P. and Barham, Paul T. “Exploiting Reality with Multicast Groups,” *IEEE Computer Graphics & Applications* (revised from appearance in the VRAIS ’95 Proceedings), September 1995, pp.38-45.

NPSNET Web Site

The full source code for NPSNET and all our research papers for the last three years can be found on our web site at the following location: http://www-npsnet.cs.nps.navy.mil/npsnet

Biography

Michael Zyda is a Professor in the Department of Computer Science at the Naval Postgraduate School, Monterey, California. Professor Zyda is also the Academic Associate and Associate Chair for Academic Affairs in that department. He has been at NPS since February of 1984. Professor Zyda's main focus in research is in the area of computer graphics, specifically the development of large-scale, networked 3D virtual environments and visual simulation systems. Professor Zyda is a member of the National Research Council Committee on Virtual Reality Research and Development. Professor Zyda is also the Senior Editor for Virtual Environments for the MIT Press quarterly PRESENCE, the journal of teleoperation and virtual environments. Professor Zyda has been active with the Symposium on Interactive 3D Graphics and was the chair of the 1990 conference, held at Snowbird, Utah and the chair of the 1995 Symposium held in Monterey, California. Professor Zyda began his career in Computer Graphics in 1973 as part of an undergraduate research group, the Senses Bureau, at the University of California, San Diego. Professor Zyda received a BA in Bioengineering from the University of California, San Diego in La Jolla in 1976, an MS in Computer Science/Neurocybernetics from the University of Massachusetts, Amherst in 1978 and a DSc in Computer Science from Washington University, St. Louis, Missouri in 1984. http://www.cs.nps.navy.mil/people/faculty/zyda