Networking for Distributed Virtual Simulation

J. Mark Pullen <mpullen@cs.gmu.edu>

Abstract

Distributed Virtual Simulation (DVS) is a new technology that extends Virtual Reality to the networked environment. It was developed for military training and is being used for civilian education and entertainment. Participants can interact over great distances in the virtual world if supported by a network that provides real-time, multicast packet communications with resource reservation for adequate performance, and satisfactory security. Requirements and techniques for real-time networking, internetted multicast, and resource reservations in support of DVS are described, and are used to define the set of missing Internet technologies that will be needed for DVS.

I. Introduction

Distributed Virtual Simulation (DVS), as developed for the U.S. Department of Defense (DoD), entails the ability for a collection of human participants to enter a virtual world containing all the essential elements of combat operations entailing air, land and sea vehicles. The individual participant can operate or ride in tanks, helicopters, fighter planes, submarines, etc.; or as a commander of computer-generated forces using such vehicles, or even as a foot soldier. The result is a system for collective training of military forces that uses no fuel, causes no training injuries (except perhaps to the ego), and avoids much of the cost associated with travel to far-away training grounds. While virtual simulation will not completely replace field training, it offers a cost-effective way to increase military readiness in times of declining defense budgets. DoD has plans to use DVS as a tool in systems engineering of new weapons, and to explore its use as a decision aid for battle commanders [1]. On the civilian side, there are many who believe DVS has untapped potential for education and for commercial success; at least one entertainment company is now developing software for arcade-style DVS.

II. Distributed Virtual Simulation

A DVS system consists of a number of simulators which use a network to exchange data about their behavior in the virtual world being simulated. The data must be exchanged in real time, with level of fidelity sufficient to permit their acceptable representation by all other participating simulators. The question "what is sufficient?" requires a detailed understanding of the purpose for which DVS is being used. For training applications it turns out that a three-dimensional rendering with visual fidelity similar to that of an arcade or home computer game is accepted well by typical users, so long as the simulator's controls are virtually the same as those of the actual tank, aircraft, etc. being simulated. The artificiality of the visual interface is quickly forgotten as the user becomes psychologically absorbed, interacting with other participants that are behaving just as they would on a real battlefield.

An important class of simulators produce semi-automated forces (SAFORs) where a human commander directs the performance of a group of artificial tanks, aircraft etc. that exist only as computer models which emit real-time information equivalent to that produced by the "real" simulators. SAFORs allow DVS to represent larger forces in order to improve the richness of the training experience for small numbers of human participants. Their quality has not reached a level such that expert humans remain convinced that they are "real" during protracted interactions, and some believe this will not be possible. Nevertheless they add to the scope of the DVS environment, particularly as many actual combat encounters are brief enough that there is little time to form such evaluations. Higher-level automated forces also are being investigated. Those interested in commercial DVS see automated participants as a powerful way to enrich the virtual environment for entertainment.

A number of major programs are underway within DoD to capitalize on results achieved within DVS to date. The Army and Navy both have programs to create training systems, while the Air Force and Marine Corps are planning similar programs. In addition DVS is being used as a systems engineering tool to evaluate alternatives for defense against the SCUD missiles that plagued U.S. forces during Operation Desert Storm, and plans call for increased DVS involvement to hold down the cost of new weapons systems and give the commanders in the field the ability to rehearse planned operations. Underlying

all of this is development of the Distributed Interactive Simulation (DIS) application-layer protocol which will allow all defense DVS systems to interoperate. A companion paper describes the DoD implementation of DVS in greater detail [2].

Interest has grown recently for commercial adaptations of the DoD DVS technology. Both commercializations of existing military systems [3] and variations for multi-player arcade and home games [4] are topics of interest; educational experiments are also ongoing. It therefore seems likely that much of the following description, expanded from the author's work on network technology for DIS [5], will also apply to the Internet of the future where Virtual Reality enables expanded learning environments, and to its commercial derivatives where it may be sold as entertainment.

III. DVS Networking Requirements

There are significant differences between the performance requirements DVS places on its supporting network, and those of more commonplace applications that are supported by TCP/IP or similar protocols. These differences can be summarized in four categories:

(1) Large-scale, *real-time* networking: because of its virtual representation of physical systems with humans "in the loop", DVS must place strict bounds on latency and jitter in packet communications among all participating simulators (which may number in tens of thousands).

(2) *Multicasting*: because it may be necessary for all participating simulators to see packets generated by all others, the network used for DVS must be able to multicast, *i.e.* deliver traffic from every simulator to all other simulators in the same virtual world. If this function is not provided by the network, each of N simulators will need to send N-1 copies of the information it generates. This means that, across the whole network, N^2-N packets must be sent where N would be sufficient. The cost of doing this, both in network capacity and simulator power, would be unacceptably large for almost all DoD exercises, and even less acceptable in civilian applications.

(3) *Resource reservation*: to achieve real-time response in a multi-user shared network, there must be a capability to reserve link bandwidth, buffers, and processor capacity. Achieving this in the Internet environment is difficult; doing so in a very large multicast network is an unsolved problem.

(4) DoD simulations often have an additional requirement for *security* in that they contain information regarding war plans or military capabilities that might be useful to a potential adversary and therefore must be protected. Network security issues are beyond the scope of this paper and will not be further addressed, but it is important that providers of network technology be aware of this firm and important requirement.

Each of the above four requirements is typically much easier to meet in a local-area network (LAN) environment than in the wide-area Internet. Real-time response is assured for any LAN that is not overloaded; multicast is an inherent capability of common LANs, and is facilitated by the fact that they typically use broadcast media, or circulate all data through every LAN node; resources on the LAN are all at one site and therefore often can be managed by configuring the attached host software; and security is much simpler because the LAN can be confined to a physically secure facility. The wide-area network (WAN) case is more difficult and contains serious unsolved problems for defense and commercial DVS, therefore the remainder of this paper will focus on the WAN case.

III.A. Real-Time Networking

The Internet protocols were originally developed in an environment where reliable transmission of data held a higher priority than the rapid, steady end-to-end transmission that characterizes real-time human-in-the-loop applications. Indeed, use of the Transmission Control Protocol (TCP) that is closely associated with the Internet provides a very high reliability, but does so in a mode that delays delivery of sizable blocks of data whenever a packet is lost, and therefore is inconsistent with real-time delivery. It is a tribute to the generality of the Internet Protocol (IP) that it has recently been used as a basis for real-time multimedia transmission (see discussion of IP Multicast below). However most TCP/IP systems remain firmly attuned to reliable data transmission; a whole new generation of networking software must be deployed to facilitate real-time networking.

The most critical parameter of a real-time network connection is its *latency*, the elapsed interval from the time a typical data segment is turned over to the sending-end application program interface (API), until the same data segment is delivered to the receiving-end application program interface. The Communications Architecture for DIS [6] specifies that network latency must be less than 100 milliseconds to

support tightly-coupled simulators (such as high-performance aircraft) and 300 milliseconds in all other cases. While there are those who feel that these numbers could be relaxed somewhat, it is clear that realism will suffer for humans involved if latency exceeds a few hundred milliseconds.

Network latency has many components: processing by host networking software, LAN queueing and transmission, send and receive site router queueing and processing, and transit time through backbone circuits and routers. Any of these components can cause latency to exceed its specified limits, therefore all must be considered when creating a network for DVS. One approach is to provide significantly more capacity in all network elements than is expected to be needed, and architect the network such that network paths among all nodes have "speed-of-light" propagation delays less than the maximum acceptable latency. Using this approach would result in poor use of tax dollars by DoD, and unacceptably high costs for commercial DVS; such a network cannot be shared among multiple applications, or even among multiple simultaneous simulation exercises The alternative, network resource reservation, is described below.

Another parameter of concern in real-time networks is *jitter* or variation in latency. The signals driving sensory output to human participants must have low jitter to sustain smooth video and audio. So long as average latency plus maximum jitter is within overall latency limits, jitter can be smoothed at the output by buffering, assuming the data is time-stamped. However, if the network limits jitter by guaranteeing both lower and upper bounds on latency, the need for smoothing may be eliminated.

III.B. Multicast Networking

As explained above, the need for multicasting to supporting networked DVS grows rapidly with increasing numbers of participating simulators. With only ten simulators, network traffic is nine times as great without multicast; with one hundred simulators, it is 9,900 times as great! Multicast is a feature of prevailing LAN technologies (Ethernet, token ring and FDDI), although it may be difficult to invoke from available APIs. The problem of establishing multicast in the wide-area DVS Internet is then to:

(1) Organize simulators into groups that can "see" (sense) each other, based on being in the same instance of the virtual world and being within sensor range of each other. This problem requires deep understanding of the objects being simulated and their characteristics, and therefore cannot be supported by generic networking technologies. It is described more fully in [2]; research into workable solutions is only beginning.

(2) Arrange LAN multicast groups at each site corresponding to (1), and use an appropriate network-layer multicast protocol and multicast address for each group.

(3) Arrange for the router (or "gateway") connecting the LAN to the WAN to understand and forward the multicast protocol.

(4) Create within the WAN a simplex spanning tree (Figure 1) from each site in every multicast group to every other site in that group. (Note that available multicast WAN protocols provide only for unidirectional flow, from the root of the tree to its leaves; however full multicast interconnection is possible by creating one tree per site in each group.)

(5) Arrange for each fork in each tree to replicate packets from its multicast group and send them toward each leaf in that tree.

(6) At the leaf routers, place the multicast packets on the LAN using an appropriate LAN multicast group address.

The above outline, simple in concept, can be difficult to implement due to the fact that no WAN multicasting protocol exists which is fully developed and widely deployed. The Internet Stream protocol, version 2 (ST) [7] has considerable history of use for DVS and includes a capability for resource reservation and is used in some production networks, but it is not widely deployed in the Internet and is still the object of ongoing efforts to create a workable standard with interoperable implementations [8]. The Internet Protocol Multicast (IP-MC) [9,10] was deployed more recently in the MBONE or virtual multicast backbone, is available more widely in the research community but is not yet supported in commercial routers (although major vendors have announced plans to do so), and has been used only minimally for DVS [11]. Each of these protocols has its adherents; the author is of the opinion that features of each need to be incorporated into the entire Internet if real-time multicast applications such as multimedia and DVS are to use the Internet successfully.

The principal difference between the ST and IP-MC protocols is the way in which they go about building the multicast tree. ST proceeds from the root toward the leaves, reserving resources as it goes (see below). It does not contain its own routing protocol, but assumes the existence of some Internet routing protocol to determine paths from root to leaves. ST can be characterized as

"sender oriented", in that the structure of the tree is initiated by the root (sender). A leaf that wants to join the tree must contact the root to be added. IP-MC puts in place a tree framework (the MBONE routers) before the multicast session is initiated; during a multicast session, a leaf that wants to join the tree sends a control message toward the root, but the leaf "twig" is added to the tree at the first node it encounters which is active in the session. This can be characterized as "receiver-oriented" multicast, and is inherently more scalable in that it avoids the problem of large numbers of leaf nodes overwhelming the root with control messages. It is however less desirable from a network security standpoint (a discussion of the reasons for this is beyond the scope of this paper).

An important aspect of multicast performance is the time for an arbitrary host to join an arbitrary multicast group, when large numbers of such joinings are taking place. This is one of a number of issues that hinge on the *control protocol* associated with the network data multicast protocol. It is typical for each network protocol to have an associated control protocol (ICMP for IP, IGMP for IP-MC, SCMP for ST). For large-scale DVS use of the Internet, the performance of joining and other control functions may be stressed severely. This is true both for DoD's aspirations toward exercises with 10,000+ objects on the battlefield, and for educational or commercial applications that may have smaller numbers of participants but involve many more simultaneous groups.

IP-MC also uses special multicast routing protocols (MOSPF and DVMRP) that recognize multicast-capable nodes. Both ST and IP-MC can be configured with paths that cross non-multicast-capable portions of the Internet by encapsulating multicast packets in IP "tunnels". This is a messy, ad-hoc solution that becomes increasingly difficult to manage for larger configurations. For effective support of DVS, multicast routing needs to become a standard feature of IP routers, removing the need for these work-around techniques.

III.C. Network Resource Reservation

It is clear that a network to support DVS, in order to provide real-time communication, must somehow reserve adequate resources along each path to be used by DVS packets, to guarantee that the path meets the end-to-end latency requirement. The key resources to be reserved are link bandwidth, node buffers, and node processor capacity. The specification associated with a requirement is called a *flowspec* and is propagated along the path by a resource reservation protocol. The flowspec typically defines bandwidth, latency, and quality of service (QoS) to be reserved in each network element along the path. (A QoS might be a code representing "packet loss not to exceed 1 in 100".) The reserving application may not always use the capacity it has reserved, in which case non-reserved traffic, if present, may use the capacity. This allows for harmonious sharing of the Internet among real-time applications and data flows that need only "best-effort" service.

While the flowspec defines the bandwidth requirement directly, node routers determine the reservations needed for memory, processor, and any other resources from the latency and QoS in the flowspec. There is a very close relationship between the data link layer mechanisms in the WAN and the ability to reserve network capacity. The most common approach applied to reservation is breaking the link data stream into fixed-size

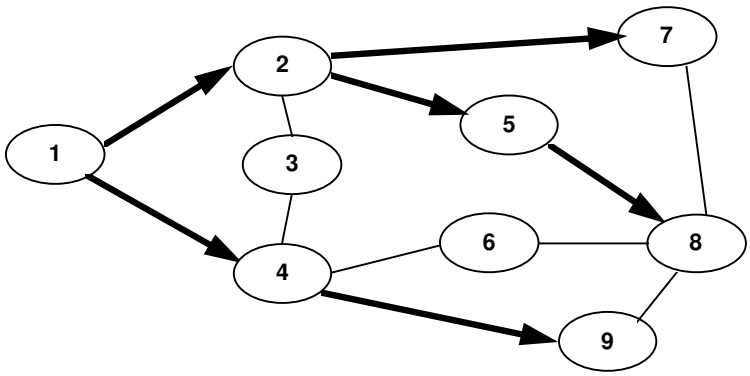

Figure 1: Wide Area Network with Multicast Tree
(Source/Root at Node 1)

slots, so that a fixed, periodic fraction of slots can be assigned to a data flow, along with an appropriate fraction of buffers, processor capacity, etc. DoD's Defense Simulation Internet (DSI) [DSI] currently uses this approach in its backbone, as does the emerging Asynchronous Transfer Mode (ATM) commercial broadband link level protocol [13].

Network resource reservation is the element that needs the most significant expansion in the Internet to support DVS. Existing capabilities for resource reservation are the Internet Stream protocol (ST) discussed above, and the Resource reSerVation Protocol (RSVP). Both ST and RSVP support resource reservation for unicast (point-to-point) data flows, but they both also are intended for use in a multicast environment. A fundamental issue where the two protocols differ lies in whether the path taken by packets is viewed as a connection that is intended to stay in place for a complete session (barring failures along the path), or as simply a route that can change dynamically at any time during the session. These viewpoints can be characterized broadly, the former being "connection-oriented", the latter "connectionless". IP has always been a connectionless protocol, therefore IP-MC and RSVP expand on that tradition, whereas ST is connection-oriented.

Because ST takes a connection-oriented approach, it reserves resources during the process of building the multicast tree. The ST flowspec is capable of identifying both desired and minimum acceptable values of bandwidth and latency, and supports a negotiation process whereby all nodes in the completed tree reserve an identical bandwidth, the best that is available in all of the multicast paths, but not more than the desired value. While this approach suffers from the scalability limitations cited earlier for source-directed multicast, ST is operating in some networks today (most notably the DSI) and has been used successfully for simultaneous multimedia conferencing and DVS applications involving tens of sites. To enhance reliability, ST contains an option for "healing" broken connection paths, even though its presumption is that paths normally do not change during a session.

RSVP [14] seeks to provide a resource reservation capability for IP and IP-MC, in cooperation with a new routing protocol (yet to be fully defined). The goal of RSVP is to accommodate a wide range of resource reservation options in the context of the connectionless Internet Protocol. Its operation requires two sets of control messages: a *PATH* message that propagates from the source to IP-MC router nodes, informing them that the path is available to reach the source; and a *RESV* message that propagates back toward the source until it finds a router that has been incorporated into the tree and can provide performance parameters good enough to meet the flowspec. In RSVP the flowspec is accompanied by a *filterspec* that designates which sources are to be received: any/all sources, a particular source fixed for the duration of the session, or a "channel switching" dynamic filter. Although RSVP includes support for admission control, it is intended to be the case that very often resource-reserved IP-MC paths to leaves are completed without involving the source; thus preserving the scalability of IP-MC. An important concept for RSVP is *soft state*, whereby the state that must be kept in routers along a reserved path is refreshed periodically by "keep-alive" control messages. The purpose of the soft state approach is that, when paths change due to IP routing updates, the resource reservation may move to the new path while the old path goes away gracefully. Also the resources are actually reserved when the receiver makes a reservation, rather than at the time the sender begins transmission, so that no reservation will exist unless it is being used.

Only one prototype implementation of RSVP exists at present. Given the advantages offered by RSVP, one would expect a great deal of interest in its deployment. However, there is an important issue that must be addressed before RSVP can become a part of the working Internet. In order to be effective, RSVP must exchange information periodically with one or more routing protocols, and provisions do not yet exist for this exchange. Also for maximum effectiveness the companion routing protocol will need to consider resource availability in its routing decisions, a criterion that is considered by current Internet routing protocols only to the extent that reduced congestion implies resource availability.

Clearly it is desirable for the Internet to support the most heterogeneous, flexible, robust approach practical for resource reservation. It is not so clear at the moment whether this involves a connection-oriented or connectionless approach, and it is even less clear when RSVP will become mature enough to serve as a mainstay of the production Internet. These are issues which need to be resolved before the Internet can support DVS.

IV. Missing Internet Technology

Based on the above overview of DVS requirements and existing Internet capabilities, it is possible to list some important capabilities which must exist if the Internet is to support DVS applications among multiple sites. These are:

Multicast Internet Protocol: A single standard multicast protocol, refined to support both sender and receiver initiation of data flows, needs to be deployed as a standard feature of every IP router.

Resource Reservation Protocol: A common means of reserving resources across all Internet routers is required. This also implies a need for standardized flowspec, admission control, and packet scheduling that while not part of the reservation protocol are needed to achieve effective resource reservation.

Compatible Routing Protocol: A routing protocol, or family of routing protocols, is required to work with the adopted multicast and resource reservation protocols.

Interface to ATM Link Layer: In the future, ATM will be the dominant broadband link layer for the Internet; therefore it is important to develop interfaces between Internet protocols and ATM systems that can take advantage of ATM's ability to reserve bandwidth and provide multicast services in the underlying communications network.

Transport Protocol: The reliable transport of TCP cannot be supported in either a multicast or a real-time environment. Most real-time applications use the User Datagram Protocol (UDP); a few use the Real-Time Protocol (RTP) which despite its name is generally applicable to audio-visual media, not to real-time simulation. The compromise between timeliness and reliability adopted by DoD's DIS protocol for DVS is to retransmit the complete state of the simulated system in every packet. This is effective but grossly inefficient. Use of a light-weight transport protocol supporting selective reliability has been proposed [15]. The author believes that a standard Internet transport protocol with selective reliability would be of great benefit to DVS. Such a protocol should assume that lost packets are rare, and provide recovery mechanisms for occasional losses, in cases where the data is critical.

Network Management for Real-time Applications: Real-time network applications are critically dependent on the network, so that the ability to manage the network for highest availability is essential. In DoD thousands of participants may sit idle when the network fails during a DVS exercise; in commercial applications, thousands of sales would be lost under equivalent circumstances. It is therefore important that all elements of the real-time network be compatible with the SNMP management protocol to permit most effective management, and that automated tools be available for fault isolation and recovery. In this regard, real-time networking is only a special case of the general need for better network management support in the Internet.

Session Protocol: Setup of multicast flows is a complex activity that is repeated by every application. The process is analogous to making a TCP connection, but is complicated greatly by the need to deal with multiple media at multiple sites. A standard Internet Session Protocol would permit sharing of supporting software and simplify the process of achieving interoperability across heterogeneous hosts, much as TCP does for unicast, best-effort data flows.

Security Architecture: Without a full discussion of network security issues, it is only possible to say here that a wide range of Internet applications are being held back by lack of an adequate treatment of security, supporting both privacy and integrity of data. This can only be achieved when a suitable security architecture is put in place and required to be implemented for participation in the Internet. The solution need not be costly for hosts with minimal security requirements, but needs to be integrated across the Internet and therefore is likely to have some impact on every IP router, if not every IP host. Solving this problem must be a high priority for the Internet community. It affects DVS and every other application.

Deployment Strategy: None of the changes listed in this section will be feasible unless the technical solutions are accompanied by a strategy for deployment where the pain of transition is offset by the gain in functionality. The Internet community therefore must plan very carefully, as any changes of the magnitude implied can only be made at very rare intervals.

If the above resembles an outline for the Internet Protocol- next generation (IP-ng), that is not accidental. The author is convinced that the time is at hand to plan and deploy a sweeping set of changes in the Internet, to complete the process of "growing up" from a research testbed in the

1970's to a globally interoperable communications infrastructure for the 1990's and beyond. Real-time services must be a part of the Internet of the future. Toward this end the needs of DVS have been put forward.

V. Conclusion

Distributed Virtual Simulation exists today as a cost-effective tool for the military. Soon it will expand to civilian education and commercial entertainment. DVS technology would be immensely more attractive if it were supported in the wide area by the Internet protocol suite. This paper has outlined the mechanisms required for internetted DVS, and described protocols that are becoming available in the Internet to supply these mechanisms, along with their shortfalls. To the extent that these shortfalls are addressed within the Internet, DVS can become part of reality in the next few years.

This paper was supported in part by the Defense Modeling and Simulation Office under contract DCA100-91-C-0033.

VI. References

[1] National Defense Magazine, special issue on simulation, Vol. 78 No. 492, November 1993

[2] D. Miller, "Network Performance Requirements for Real-Time Distributed Simulation", Proceedings of International Networking Conference, Prague, Czech Republic, June 1994

[3] S. Seidensticker, "Distributed Simulation: A View From The Future", electronic mail white paper, seiden@netcom.com, 1994

[4] M. Lawlor, "Firms Take Technology from Battle Simulator to TV Screen", AFCEA Signal Magazine, Vol. 48 No.8, April 1994

[5] S. Symington, D. Wood, and J.M. Pullen, "Modeling and Simulation Requirements for IPng", electronic mail white paper, mpullen@cs.gmu.edu, February 1994

[6] Distributed Interactive Simulation Working Group, "Communications Architecture for DIS", University of Central Florida, Orlando, Florida, 1993

[7] C. Topolcic, "Experimental Internet Stream Protocol, Version 2 (ST-II)", RFC1190, Internet Engineering Task Force, 1990

[8] L. Delgrossi, R.G. Herrtwich, and F.O. Hoffman, "An Implementation of ST-II for the Heidelberg Transport System", IBM European Networking Center, Heidelberg, Germany, 1993

[9] S. Deering, "Multicast Routing in a Datagram Network", Stanford University, Stanford, California, December 1991

[10] S. Deering, "Host Extensions for IP Multicasting", RFC1112, Internet Engineering Task Force, 1989

[11] M. R. Macedonia, D. R. Pratt, and M. J. Zyda, "A Network Architecture for Large Scale Virtual Environments", US Naval postgraduate School, Monterey, California, 1994

[12] R. Wetzel, "The Defense Simulation Internet: User's Guide", BBN Systems and Technologies, Cambridge, Massachusetts, October 1992

[13] IEEE Communications Magazine, special issue on High Speed ATM Switching, Vol. 7 No. 2, March 1993

[14] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", IEEE Network Magazine, Vol. 7 No. 9, September 1993

[15] D. Cohen, "DIS Next Generation PDU", Perceptronics, Inc., Los Angeles, CA, March 1994

Author Information

J. Mark Pullen is an Associate Professor of Computer Science at George Mason University in Fairfax, Virginia, USA, where he teaches networking and distributed applications and performs research as a member of the Center for Excellence in Command, Control, Communications and Intelligence. He holds Bachelor's and Master's degrees in Electrical Engineering from West Virginia University and the Doctor of Science degree from The George Washington University, and is a Fellow of the IEEE. Prior to joining the George Mason faculty he held several positions at the US Defense Advanced Research Projects Agency (DARPA, now ARPA) including Program Manager for Advanced Networking and Program Director for Defense Distributed Simulation; in the latter capacity he put in place a worldwide network for distributed simulation that provides initial services along the lines described in this paper.