

Enabling Real-Time Distributed Virtual Simulation over the Internet Using Host-based Overlay Multicast

Dennis M. Moen & J. Mark Pullen

George Mason University
{dmoen,mpullen}@gmu.edu

Abstract

Many-to-many multicast transmission is an essential network capability for scalable distributed simulation. The many open issues that make network-layer multicast impractical over the Internet today are likely to continue to be insurmountable and to prevent deployment in a manner that will meet the Quality of Service (QoS) and many-to-many multicast needs of real-time distributed virtual simulation (RT-DVS). We propose an alternative approach, whereby the IP multicast function is relocated to the simulation host computer and also used to provide for QoS enabled services on a priority basis. This paper explores the concept of such a host-based multicast system with QoS and its suitability for RT-DVS. The paper defines the basic requirements of a host-based multicast protocol to support RT-DVS and proposes a quantitative model for evaluation of system alternatives.

Keywords: distributed virtual simulation, multicast networking, overlay multicast

1. Introduction

Many-to-many multicast transmission is an essential network capability for scalable distributed simulation because the more common unicast approach does not scale well with many-to-many traffic. Network-layer Internet Protocol (IP) multicast over the Internet was introduced in 1993 [1] but still is not widely deployed. There have been inherent disincentives for public carriers to implement IP multicast in their networks, the largest being financial. The carriers do not currently have a business model that lets them make money on multicast service in an open network which crosses multiple carrier domains. Public carriers invoice based on fixed or burst levels of traffic, whereas multicasting is designed to reduce offered traffic load for group communication. While the Internet Engineering Task Force (IETF) is exploring one-to-many multicast over the Internet on the premise that there is growing support for this service, it is unlikely that many-to-many multicast over the Internet will ever materialize. There are also concerns for broad deployment of IP

multicast related to reliability, congestion control, network-to-network policy sharing, and scalability [2].

IP multicast's lack of commercial viability has led to a growing interest in *end system multicast*, where all multicast related functionality is implemented among the application computers as an overlay structure. In this overlay, all functionality of multicast including group management and packet replication is performed in the end system (or *host*, in Internet terminology). Such end system approaches include optimization of the overlay by adapting to network dynamics and are able to take application performance into consideration [2].

Performance of the end system approach has been studied for the class of applications supporting audio and video conferencing [2]. Results of the studies are highly favorable for the end system multicast approach.

Audio and video conferencing applications have very stringent performance requirements for delay and throughput, similar to the requirements of real-time distributed virtual simulations. The distributed simulations operate with human participants who expect to experience a virtual presentation in real-time [3], not unlike the requirements of audio or video conferencing. A major difference between the conferencing and simulation environments is that, in conferencing, jitter is managed in the end system application layer, generally by buffering incoming traffic to create a constant delay. In RT-DVS, adding delay is problematic so that jitter must be managed real-time and inherently be part of the service offered to the application.

End system multicast provides an attractive alternative to network based multicast by allowing the distributed simulation to become independent of network based multicast. The case for this approach also can be based on the historical end-to-end argument [4] that a functionality should be (a) pushed to higher layers if possible, (b) unless implementing it at the lower layer can achieve large performance benefit that outweighs the cost of additional complexity at the lower layer. For example, in the Internet Protocol Suite ordered, reliable delivery and congestion control are implemented in the Transmission Control Protocol (TCP), which operates in the end systems, not in the lower layer IP.

to appear in

Proceedings of the IEEE/ACM Distributed Simulation-Real Time Applications Symposium 2003

End system multicast for real-time distributed simulations will allow the continued use of open protocols at lower layers, implemented across the Internet. This will support RT-DVS in a way that is not dependent on private networks or consistency of network policy implementation across multiple carriers, which are two major limitations to wide-area deployment of RT-DVS today.

2. End System Multicast Definition

In this section we describe how end system multicast works and relate it to IP multicast as implemented today. Distance Vector Multicast Routing Protocol (DVMRP) is the oldest multicast routing protocol and was first defined in RFC 1075 [5]. Using DVMRP, data is delivered from the source to the receivers using an IP multicast tree of the shortest paths from each receiver to the source. Wherever possible, traffic for group members shares the same network links. Figure 1 presents an IP multicast tree constructed using DVMRP. Routers A and B receive a single copy of each packet and forward it along multiple interfaces. However, only one copy of the packet is sent over any physical link. Packet delay is the same to all receivers as though packets were sent directly by unicast, but network loading is decreased.

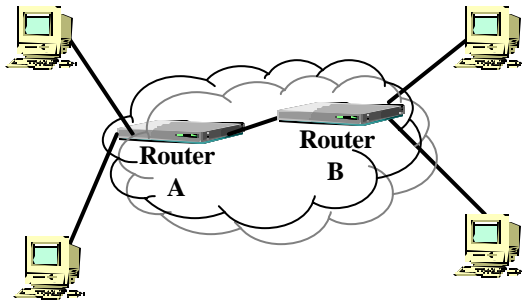


Figure 1. IP Multicast tree using DVMRP

We can expand the representation of the multicast function from this simple example to a complete graph (Fig. 2) where every pair of nodes is the set of ends of an edge representing the virtual flow of packets [6]. DVMRP routing then requires solving the minimum spanning tree problem for this complete graph. If we now remove the multicast function from the routers A and B in the network and build a new tree based only on the end host systems, we get a spanning tree as presented in Figure 3. The underlying physical path of the packets now is represented as in Figure 4. The result is that the traffic crosses some of the physical paths twice with a small amount of delay added as a result. The End System multicast concept can be generalized to include nodes at the edge of a network,

where proxies can use the LAN multicast functionality [7]. Such a proxy also can act as a router, forwarding packets on behalf of other nodes (Fig. 5).

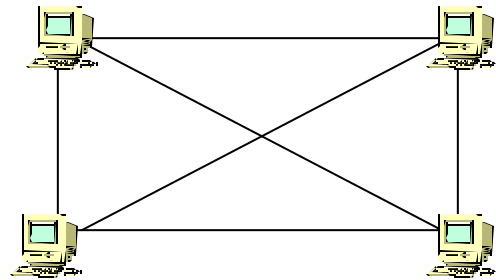


Figure 2. Complete Graph



Figure 3. Spanning Tree Graph

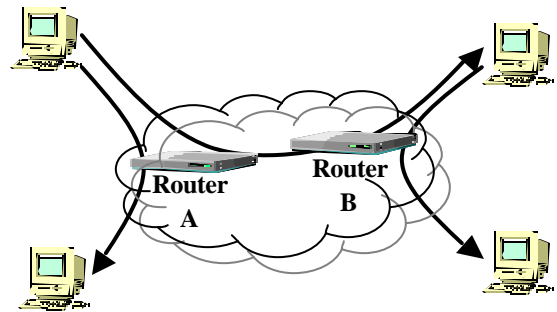


Figure 4. Physical path of the packets

The tree approach to multicast overlay is simple and efficient, resulting in a single path between any pair of nodes in the overlay network. BTP, HMTP, and Yoid [8] are examples of experimental tree-based multicast overlay protocols. While the simplicity of these protocols is desirable, they are vulnerable to single link failure potentially resulting in the partitioning of group membership [9]. For robustness, the tree class of protocols requires mechanisms to protect from loss of a link.

For highly dynamic environments with QoS requirements such as that of RT-DVS, it may be more desirable to consider mesh-based overlay multicast protocols. Overlay protocols such as NARADA and Hypercast [8] provide mechanisms for redundant

to appear in

Proceedings of the IEEE/ACM Distributed Simulation-Real Time Applications Symposium 2003

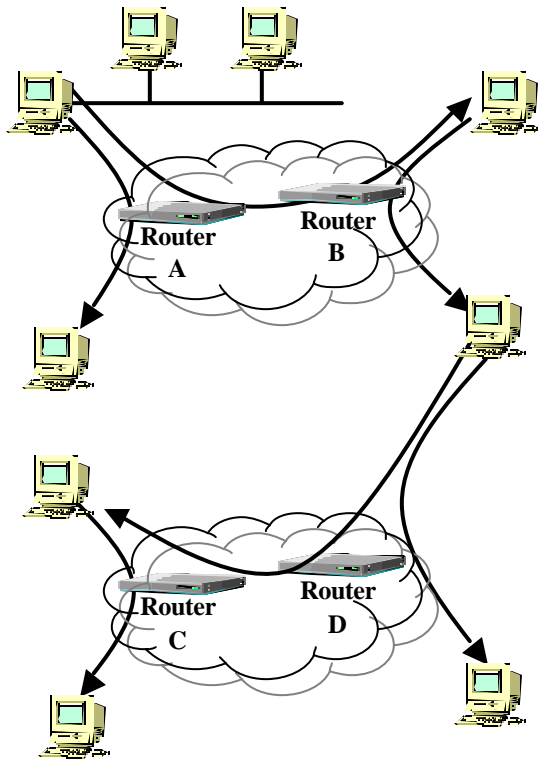


Figure 5. End system proxy nodes

connections between group members in the overlay, offering a protection against single link failure. However, these protocols are more complex than tree-based protocols because they must include mechanisms to protect against routing loops between group members [8].

3. Performance Understanding in a Complex Internet

Widely deploying RT-DVS across many organizations and a large number of applications implies networking across the multitude of Autonomous Systems (AS) in the Internet. Each AS operates under separately managed network routing policies with no guarantees that routing policies are reflected to any other AS. Consequently, network efficiencies and performance will follow the least common denominator and may very well be inconsistent.

Each AS network manager defines a routing policy that specifies how routes are selected and advertised to each neighboring AS. Generally, these managers implement routing schemes that favor certain AS route destinations, encourage other AS's to favor certain Network Access Points (NAP), or advertise a preference on how to be reached across multiple NAPs [10]. The net effect is that there is no guarantee of consistent service for delay or available bandwidth across the Internet.

RT-DVS applications require a predictable network performance to meet stringent requirements for network

capacity, delay, and jitter [11, 12]. While the Internet protocols MPLS and DiffServ offer managed QoS that potentially meet these requirements, they also are managed independently in each AS and therefore do not provide an Internet-wide solution. MPLS also is an end-to-end service where all intervening WAN network elements need to understand and respond consistently to the same MPLS labels. Thousands of competitive Internet Service Providers (ISP) make up the Internet along with a large number of private networks; therefore, it is highly unlikely that MPLS will ever achieve an end-to-end consistency supportive of a common MPLS infrastructure [13].

For end-to-end system performance, there still remains a higher-level requirement to provide a managed QoS service across the independent domains. To achieve this, every AS manager must have an incentive to maximize performance across the managed domain. The choices made by each manager must not be hindered by any other AS in order for the end-to-end system application to achieve desired results across the network. To overcome this limitation, Chu has proposed a self-organizing multicast protocol for end-system multicast called NARADA [7]. This protocol attempts to adapt to both latency and available bandwidth by performing active measurements across paths established by the NARADA protocol. With this independence of AS, an overlay multicast protocol is able to take advantage of the service offered by any MPLS or DiffServ managed AS without even the necessity of knowing that the AS is using this underlying technology. This approach would provide the network independence necessary for a set of real-time distributed simulations to interact effectively within the confines of the network overlay.

While NARADA experimental results show a way to provide capabilities for multicast performance based on bandwidth and latency, RT-DVS applications also require consideration for management of jitter. In addition, there are a number of other factors that need consideration. For example, time synchronization is extremely important. The applications also must be "network aware," e.g. feedback mechanisms for QoS must be part of the overlay service, in order to support an end-to-end managed QoS service.

4. Value of Overlay Multicast for RT-DVS

RT-DVS requires specific delay bounds to support the delivery of real-time, interactive visual and audio information at human response times. The environment can be described as a multiparty collaborative environment running multimedia applications. The underlying networking environment needs to support a large number of participants that dynamically join and leave the communicating group across the myriad of public and private networks that make up the Internet. Because of this

nature of the Internet, it is not able to deliver the necessary end-to-end QoS required for RT-DVS. It therefore appears likely that a RT-DVS application running an end-to-end multicast protocol will improve the ability to take advantage of underlying QoS protocols across the Internet. The overlay approach could take advantage of commercial efforts to improve QoS without the need to rely on specific deployment on a specific network of QoS mechanisms.

This approach also may help to expand the RT-DVS user base since the concept could extend to individual personal computers (PCs) acting as relay nodes, enabling low-cost end-to-end multicast schema. The omnipresence of low cost, very fast processor PCs as well as the growing presence of inexpensive broadband access has the potential to make this a reality.

This strategy also will support the efforts to implement Web-based strategies to bring together hundreds of legacy applications in the current generation of defense related modeling and simulation environments [14]. These efforts have resulted in the formation of the Extensible Modeling and Simulation Framework (XMSF) that describes requirements for a web-based approach to support future modeling and simulation environments. This framework has the objective of providing an environment that will enable simulations to interact directly and to be scaleable over a highly distributed network through compatibility with the web framework and technologies.

The XMSF strategy is to take advantage of XML-oriented Web services to realize the XML potential for application-to-application interoperability that leverages lower layer network services of IP, HTTP, proxy services, and simplicity of the Internet networking. Under the XML-oriented model, the flow of multicast based content for the RT-DVS must support delivery of the XML encapsulated content with the efficiencies and QoS described earlier. Since web based models have traditionally been session oriented, a new strategy is required to provide for the QoS demands of RT-DVS. An overlay multicast approach can provide this capability.

5. Network Services for RT-DVS

The RT-DVS networking environment is very complex and is not easily supported in a unicast Internet environment. Figure 6 depicts how group communication over unicast networks becomes complex, even for a small number of users. Simulation data sources can be receivers and vice versa, while both can act as "virtual" nodes or overlay routers in the network. Thus, a small number of users can precipitate a very complex multi-path problem.

High-quality RT-DVS requires that QoS be specified or negotiated for capacity, latency, jitter, and packet loss (in a statistical sense) on an end-to-end basis, or between applications. In general terms, QoS can be specified or

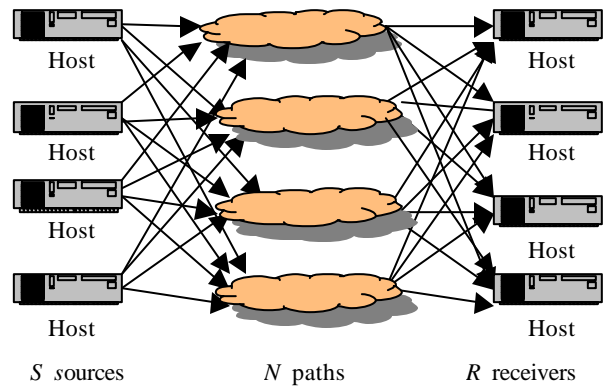


Figure 6. Many-to-Many Multicast

negotiated within private networks or individual ISP (Internet Service Provider) networks, but this is not possible today across the broader Internet. Further, no known approach will support Internet wide QoS negotiation. Adding to the difficulty is the fact that achieving QoS objectives implies tradeoffs. For example, reliability and latency are correlated. Increased reliability implies greater latency with the result that it is impossible to have fully reliable, real-time multicast. Thus, if RT-DVS is able to function more effectively with reliable transport for some data, it is necessary to specify reliability in the format of mixed, selectively reliable/real-time and fully reliable/near-real-time capability.

Classic one-to-many multicast to support video or audio streaming generally can be implemented with minimal spanning tree routing. Current routing protocols use the approach explained above. Because RT-DVS demands a higher level of QoS and many-to-many multicast, the new problem becomes one of dynamically extending the multicast tree problem in an overlay architecture because of the presence of many Autonomous Systems, as described above. There are two special cases that result in the problem being dynamic:

1. Network events: disruption, congestion (QoS requirements not being met), and/or spare capacity re-allocation optimization may increase the cost of the edges in the current tree or decrease that of the edges not currently in the tree as a result of multicast bandwidth re-allocation since the current tree was computed. This results in the need for dynamic re-calculation of the tree.
2. The RT-DVS environment can have a very large multicast group with highly dynamic characteristics. Objects in the virtual simulation may join and leave many multicast groups frequently as the scenarios of the simulations are carried out. This dynamic group characteristic is likely to result in the frequent need to re-calculate the multicast tree to re-allocate bandwidth

or generate new paths or new flows within paths to support the changes.

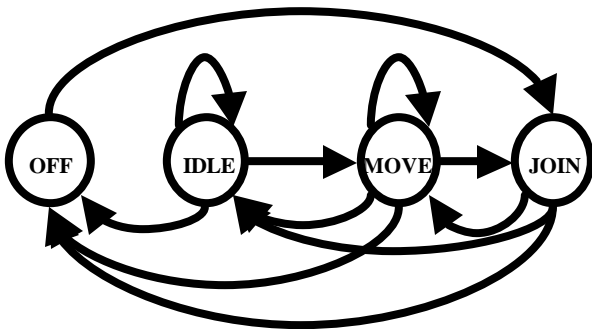
6. Performance Modeling Considerations

To summarize the above: The RT-DVS is dependent on real-time response and predictable behavior in order for the end systems to interact with the physical world within specific delay bounds and present multimedia data on a real-time basis. Users deployed across the Internet and/or Intranets require low latency, including stringent jitter requirements and high bandwidth demands. In this section, we address factors to be considered in modeling system performance in order to develop an overlay multicast architecture for the RT-DVS environment.

6.1 Traffic Characterization

A traffic model with appropriate metrics is required to describe the behavior of the RT-DVS. These metrics must take into account packet error and loss rates, throughput, latency and jitter, and path flows including blocking. Generally, this is complicated by the nature of the application as the traffic load is specific to a simulation scenario. To facilitate studies of traffic load behavior of protocols supporting the RT-DVS, a traffic model has been created [12]. This model is based on a 4state process to represent status of a simulation object. A state diagram for this model is presented in Figure 7. The approach represents a continuous-time stochastic process with a discrete state space. Two rules are applied to state change:

1. If the system jumps to state i , it stays in state i , an exponentially distributed time with a mean m
2. If the system leaves state i , it jumps to state j with probability $p_{ij}(j \neq i)$ where $\sum_j p_{ij} = 1$ for all $i \in I$.



OFF: No Traffic generated
 IDLE: Object stable in a multicast group
 MOVE: Object is moving within multicast group
 JOIN: Object joins a new multicast group

Figure 7. Traffic State-Transition Diagram

6.2 Demand Characterization

A model for RT-DVS resource demand is required to support development of scalable solutions. The model must include bandwidth assignment at each server and dimensioning/quantifying the load at the servers according to a multi-class session traffic model. The scalability question to be addressed is how to assign the bandwidth of the link(s) that connects the server to the Internet such that acceptable service levels are achieved and maintained at the lowest cost. The model approach requires implementation of an open multi-class queueing network model where C is the number of classes (flows) and I_c ($c = 1, 2, 3, \dots, C$) is the arrival rate of class (flow) c . Based on this model, service demand in terms of network capacity can be calculated.

6.3 Path Characterization

The overlay approach requires discovery of valid network paths and associated QoS. The objective is to define a minimal network overlay while grouping flows (multi-cast aggregation edges of a graph) such that a given grade of service (QoS) is maintained. This implies a need for an ability to scale active end-to-end measurements to estimate path costs such as latency, jitter, and network capacity.

6.4 Path Convergence

The approach requires dynamic overlay network optimization to support a very robust environment, requiring many-to-many multicast with multicast group members joining and leaving the groups real-time. The problem is to dynamically find a network that connects a finite set of points in a metric space with the shortest possible length. It is also important to consider the time required for network optimization to occur for the real-time applications. This may prove difficult, because end systems typically have limited topological information in which to construct good overlay architectures (paths). The problem is made more complex in that the approach could allow insertion of additional nodes (see example in Figure 5 above) to achieve optimization. This problem is more commonly known as the Steiner problem [15, 16]. The general problem can be defined as:

Given network $G = (V, E)$, node set V , edge set E , edge capacities c_{ij} , source s and set of receivers $R \cap N - \{s\}$, any multicast flow is identified with a collection of directed trees $\{T_1, T_2, \dots, T_i\}$ rooted at s and reaching all nodes in R and associated flows $\{x_1, x_2, \dots, x_i\}$ with constraints:

to appear in

Proceedings of the IEEE/ACM Distributed Simulation-Real Time Applications Symposium 2003

$$\sum_{(1 \leq k \leq l)} f(i,j,T_k) * x_k \leq c_{ij} \quad \forall (i,j) \in E$$

$$f(i,j,T_k) = \begin{cases} 1 & \text{If arc (i,j) appears in Tree } T_k, \\ 0 & \text{otherwise} \end{cases}$$

This multicast flow problem involves sending flow from a source to a set of destinations or receivers. A directed Steiner tree rooted at the source and reaching all receivers can be used as the basic unit of flow for the multicast network flow. A multicast flow is then defined as a collection of flows along directed trees obeying the capacity constraints for each edge in a path.

7. Future work

We have presented a case for use of overlay multicast to achieve lower cost and to expand the user base for RT-DVS by enabling multicasting, potentially including QoS on a priority basis, using end-system overlay. A number of activities are required to bring this concept of end-to-end system multicast for RT-DVS into reality. At this point it is important to gain the interest of the RT-DVS community such that a standards based approach to middleware software development is feasible. Our research strategy is to continue expansion of the definition of the concept, and define through analysis and modeling, the value of the concept. The proposed next steps are:

- Develop the network service requirements for the RT-DVS community such that an overlay; end-to-end approach for QoS based many-to-many multicast can be defined.
- Based on the RT-DVS network service requirements, develop an architecture for an end-to-end, managed QoS, many-to-many multicasting mechanism that will support the stringent requirements of RT-DVS.
- Develop an analytical model and a simulation model to support an analysis of the architecture with measures for QoS. The model should include the four considerations described above for traffic, host demand, path characterization and path convergence. Test the architecture for scalability under multiple scenarios. These scenarios should be designed to reflect variations in network parameters such as available bandwidth and path dynamics as well as traffic and host demand.
- Confirm the utility of the general approach by prototyping repeatedly, at increasing levels of sophistication and utility, and applying emerging techniques for effective management of the overlay network [17].

8. Conclusion

We have shown that end system overlay multicast has great promise as a network architecture to meet the performance requirements of RT-DVS. Traffic efficiency, latency, and jitter are critical network parameters in the deployment of RT-DVS. Implementing a multicast strategy that maintains RT-DVS independence of the underlying network architecture that at the same time takes advantage of Internet QoS implementations appears to be a reachable goal. The RT-DVS community needs to be engaged in defining requirements and solutions to make this potential into reality. We are proceeding with performance modeling for the approach and prototyping to demonstrate results.

Acknowledgement

This work was supported in part by the US Defense Modeling and Simulation Office.

References

1. Brutzman, D. and Macedonia, M., "Mbone Provides Audio and Video Across the Internet," *IEEE Computer Magazine*, pp. 30-36, April 1994
2. Chu, Yang-hua, S. G. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture," *Proceedings of ACM, SIGCOMM2001*, pp.55-67, August 2001.
3. Pullen, J., "Reliable Multicast Network Transport for Distributed Virtual Simulation," *Proceedings of the 1999 IEEE Workshop on Distributed Simulation and Real-Time Applications*, 1999.
4. Saltzer, I., D. Reed, and D Clark. "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, 2(4), pp.277-288, 1984.
5. Miller, Kenneth C., *Multicast Networking and Applications*, Addison-Wesley Longman, 1999, pp.36.
6. Cook, William I., William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*, John Wiley & Sons, 1998, pp.10.
7. Chu, Yang-hua, S. G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," *Proceedings of the ACM Sigmetrics*, June 2000.
8. Wang, Wenjie, David Helder, Sughi Jamin, and Lixia Zhang, "Overlay Optimizations for End-host Multicast," *Proceedings of the ACM NGC 2002*, pp. 154-161.
9. Garcia-Luna-Aceves, I., Daravanan Balasubramanian, and Ramesh Balakrishnan, "A Scalable and Fault-Tolerant Architecture for Internet Multicasting Using Meshes," *Proceedings of the ACM NGC 2002*, pp.87-94.
10. Savage, Stefan, Andy Collins, Eric Hoffman, John Snell, and Thomas Anderson, "The End-to-End Effects of Internet Path Selection," *Proceedings of the ACM SIGCOMM 1999*, pp. 289-299.

to appear in

Proceedings of the IEEE/ACM Distributed Simulation-Real Time Applications Symposium 2003

11. Zhao, Hui, and Nicolas D. Georganas, "HLA Real-Time Extension," *Proceedings of the Fifth IEEE Workshop on Distributed Simulation and Real-Time Applications*, 2001, pp. 12-21.
12. Moen, D. and J. Pullen, "A Performance Measurement Approach for the Selectively Reliable Multicast Protocol for Distributed Simulation," *Proceedings of the Fifth IEEE Workshop on Distributed Simulation and Real-Time Applications*, 2001, pp.30-34.
13. Mier, Edwin E. and Marc Frigo, "MPLS: The Making of VWANsT," *Business Communications Review*, May 2002, pp.46.
14. Brutzman, D., M. Zyda, M., J.M. Pullen, and K.L. Morse, *Extensible Modeling and Simulation Framework (XMSF): Challenges for Web-Based Modeling and Simulation*, US Naval Postgraduate School, 2002.
15. Cieslik, Dietmar, *Steiner Minimal Trees*, Kluwer Academic Publishers, 1998.
16. Ivanov, Alexandr O., and Alexei A. Tuzhilin, *Minimal Networks: The Steiner Problem and Its Generalizations*, CRC Press, Inc., 1994.
17. Simon, R., J. Pullen and W. Chang, "An Agent Architecture for Composable Network Service Support of Distributed Simulation Systems", to appear in *Proceedings of the Seventh IEEE Workshop on Distributed Simulation and Real-Time Applications*

AUTHOR BIOGRAPHIES

Dennis Moen received his M.S. degree in electrical engineering from the University of Arizona. He is a licensed Professional Engineer, and is pursuing a PhD in Information Technology at George Mason University. He joined Lockheed Martin Corporation in 2002 as a senior staff member in survivable network architectures and design. His research interests are network performance modeling and adaptive Internet protocols.

J. Mark Pullen is Professor of Computer Science at George Mason University, where he heads the Networking and Simulation Laboratory in the C³I Center. He holds BSEE and MSEE degrees from West Virginia University, and the Doctor of Science in Computer Science from the George Washington University. He is a licensed Professional Engineer, Fellow of the IEEE, and Fellow of the ACM. Dr. Pullen teaches courses in computer networking and has active research in networking for distributed virtual simulation and networked multimedia tools for distance education.