

Performance Issues for the Light-Weight RTI

J. Mark Pullen and Nagesh Kakarlamudi
Department of Computer Science and C³I Center
George Mason University
Fairfax, VA 22030
{mpullen,nkakarla}@gmu.edu

Keywords:

HLA, RTI, selectively reliable transmission protocol, reliable multicast

ABSTRACT: *The Light-Weight RTI (LWRTI) was developed at GMU as an extension of our work in communication architectures for distributed simulation (Dual-Mode Multicast) and transport protocols for the large multicast environment (Selective Reliable Transmission Protocol or SRTP). This paper reports on our efforts to understand performance issues surrounding the LWRTI. One set of issues, in the area of throughput, relates to tuning the LWRTI software and identifying structural questions in the HLA that may constrain its performance, which initially was somewhat disappointing. The other set of issues relates to efficiency of the Hop-Hierarchical Multicast Logging (HHML) used by SRTP to configure dynamic multicast logging such that reliable transport is provided efficiently in a complex network, for those object attributes which are most effectively dealt with by reliable multicast transmission. We report our findings in each of these areas, with conclusions regarding the potential effectiveness of SRTP and the LWRTI in a large, dynamically configured multicast network.*

1. Introduction

This paper reports on progress in our effort to understand the new DoD High Level Architecture (HLA) as applied to distributed virtual simulation, by creating and using a light-weight Run Time Infrastructure (LWRTI). At George Mason University (GMU) we have been engaged for several years in experimental work in areas related to scale-up of Distributed Interactive Simulation (DIS) to large-scale battlefield simulations (tens to thousands of active objects). As a result we have created prototypes for base technologies fundamental to the communications aspects of the RTI. The light-weight RTI is therefore a transitional effort which takes what we know about DIS and uses it to support distributed virtual simulation within the context of the HLA. We first reported on this work in [1]. Our starting points for this effort were the Selectively Reliable Transmission Protocol (SRTP), the Dual-Mode Multicast architecture (DMMC), and the definition of the HLA. Our LWRTI supports only distributed virtual simulation, extending the basic DIS paradigm to the realm of HLA. We have been able to support a small DIS-like federation with the LWRTI, but we have found its performance somewhat disappointing. Also we are engaged in expanding the underlying SRTP multicast communications support for the LWRTI, as described in [2] and [3]. This process has led us to consider models for improved congestion avoidance performance of the reliable multicast provided by mode

of SRTP, which we believe will have a serious impact on overall network performance in a large multicast environment. We report here the results of our investigations of these performance issues, which have serious impact on the usability of the LWRTI and, we believe, many also be of significant concern to implementers of other RTIs in the future.

2. Previous and related work

2.1 SRTP

The Selectively Reliable Transmission Protocol was developed at George Mason University to support large scale DIS. Our inspiration for using selective reliability to support the compression of entity state information using a reference/offset scheme came from Danny Cohen's work with the DIS community [4,5,6]. SRTP, used with the User Datagram Protocol (UDP), provides a selectively reliable transport service that embodies a tradeoff between minimal latency and reliability. SRTP exploits the known properties of distributed simulation traffic, reducing network capacity requirements by transmitting, using real-time best-effort multicast, only those data elements, such as position, that change frequently. Those data that change rarely, or not at all, are transmitted reliably one time. This approach follows from the fact that transmission parameters low latency and high reliability, most often seen as alternatives, are

in fact end points in a range of options. SRTP seeks to operate between the two end points [2].

To date the principal transport protocols used for distributed simulation have been the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP provides reliable data transport but is not appropriate for large scale multicast communications because of the phenomenon known as "ACK implosion". TCP requires that a receiver send a positive acknowledgement (ACK) in response to data segments. If the multicast group is large, this control traffic can create enough congestion that the transmission of future data segments is impaired severely. As a result broadcast and multicast virtual simulations typically use UDP, which as a best effort transport protocol does not use ACKs. However, this forces each application to provide for reliability on its own when it is required. SRTP provides a standardized set of communication services than can be combined with UDP to provide reliable transmission of data that change rarely, intermixed with best-effort transmission of frequently-changing data. It also provides relevance filtering needed for multicast group assignment. The result is a considerable reduction in network capacity requirements to support distributed simulation, which we exploit in the LWRTI. In [1] we show that, for a representative DIS case, SRTP can result in up to 83% reduction in bits per second transmitted.

SRTP API		
Relevance Filtering by Category and MC Group		
Mode 0: Best-Effort Multicast Service	Mode 1: Reliable Multicast Service	Mode 2: Reliable Datagram Service

Figure 1: SRTP Functions

SRTP provides relevance filtering and three transmission modes: mode 0 for best-effort multicast, mode 1 for receiver-reliable multicast, and mode 2 for reliable (unicast) datagram service. Modes 0 and 1 were designed to facilitate the transmission of entity state information, while mode 2 was designed for the transmission of dynamic interactions between pairs of entities without the overhead of TCP connection management. Figure 1 summarizes the functions of SRTP.

2.2 Dual-Mode Multicast

We have been investigating distributed simulation networking approaches for the synthetic battlefield that are able to support large numbers of active objects (approaching 100,000) on the battlefield. For economical implementation, this problem requires use of multicast networks to reduce bandwidth requirements and simulator input of a DIS exercise or an HLA federation. The multicasting problem at this level, known as the "large scale multicast environment" [7], requires highly scalable protocols to work. To date the most popular method of assigning objects to multicast groups has been interest management where the simulation is segmented into groups by geographic region.

An approach that we have investigated, called Dual Mode Multicast (DMMC) [8,9], provides a scalable solution for distributed simulation. It utilizes one (or a few) exercise-wide multicast group(s) on the WAN and an interest management scheme to select at each simulator from a large number of possible multicast groups on the LAN. The basic premise of DMMC is that, because it is not possible to know in advance which simulators will interact in a practical exercise, WAN backbone capacity must be provisioned sufficient to support the maximum data flow that can be generated by the simulators between any two sites. This being so, there is no additional cost if the data packets needed to depict the behavior of each simulator are sent in a single WAN multicast group that distributes them to each site.

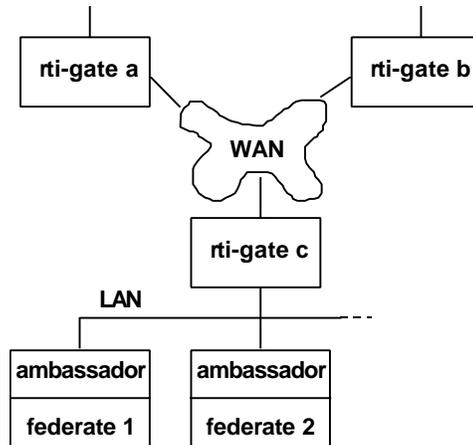


Figure 2: LWRTI Architecture (one federation)

DMMC is implemented by having a filtering process at an Application Gateway (AG) on each LAN, which (1) forwards all PDUs generated locally to an exercise-level multicast group; and (2) processes all PDUs coming from the rest of the exercise, removing or filtering any PDUs

that are not needed locally, and multicasting those remaining PDUs to the local simulators. As shown in Figure 2, we have adapted the DMMC concept to the LWRTI by implementing the AG in one RTI gateway (rti-gate) per LAN.

2.3 HLA and the RTI

The DoD Modeling and Simulation (M&S) Master Plan [10] established a goal to develop a common technical framework for M&S. The resulting High Level Architecture (HLA) is intended to provide interoperability and standardized infrastructure where possible, making live, virtual and constructive simulations work together, despite, for example, heterogeneous time management mechanisms. Services necessary to support the HLA are provided by a single logical component, called Run Time Infrastructure (RTI), which supports execution, object and time management, and data exchange. We have chosen to limit our LWRTI to real-time, virtual distributed simulation. While this necessarily limits the scope of simulations it can support, we believe we need to learn to support this important class of simulations well. To that end we have performed the experimental evaluation of the LWRTI described below. This has caused us to pay particular attention to the HLA Interface Specifications, which provides a specification for the functional interfaces between federates and the RTI.

The HLA RTI is the software implementation of the Interface Specification [11]. The services provided by the RTI are divided into six groups: Federation Management, Declaration Management (DM), Object Management, Ownership Management, Time Management and Data Distribution Management (DDM). The RTI is in part intended to increase the scalability of an exercise (when compared to the DIS protocol), by reducing the data set required to be processed by the receiving federate and the message traffic over the network. Each federate subscribes and publishes the corresponding object and interaction classes through the DM, specifying if necessary the attributes it is concerned about. The published classes determine to which class membership of objects that the federate is going to register, the attributes it is going to provide to the federation, and the attributes it may acquire through the Ownership Management services. This information is also used by the RTI to control the flow of data among the federates by performing class-based filtering. A federate which subscribes to the attribute values of an object class will receive all updates of the specified attribute values for the registered objects of that class which are "in scope".

The DM provides a class-based filtering mechanism while the DDM provides a more complex value-based filtering mechanism, based on the values of the attributes [12]. The federates express their interest in sending or receiving data in a multidimensional coordinate system, called a routing space, by defining regions. A region is a set of extents, which are described by a set of ranges in the dimensions of the routing space. A region can be a rather complex area defined by the union of several hyper-rectangles in the space. The DDM can be used in several ways: specifying an update region associated with a particular object; specifying a subscription region for a given object or interaction class; or modifying the region according to the changes of object characteristics during the simulation.

The regions are used by the RTI to deliver attribute updates and interactions. For an object, the attributes will be routed from the sender to the federates which have subscribed to these attributes for the class (or a super-class) of this object and which have specified a subscription region which overlaps the update region of the attributes. Before delivering any attribute, the RTI must inform the federate that a new object is in-scope, and use the Discover Object service. When the subscription regions of a receiving federate no longer overlap with any update region of the object, the RTI tells this federate to remove the out-of-scope object.

2.4 The Light-Weight RTI

The light-weight RTI focuses on Declaration Management and the Data Distribution Management issues necessary to support distributed virtual simulation. Therefore several services defined under the HLA have not been implemented. We started with a minimal level of complexity, omitting several groups of services that we believe are less important in the virtual simulation domain: Ownership Management and Time Management functionalities have been completely omitted because they were not needed for virtual simulation. Similarly, in order to support a federation execution only a few Federation Management services are necessary. Federates only need to create, join, leave and destroy the federation. Therefore, pausing and saving the exercise were not implemented in our version. We did implement most Object Management services, however without Time Management, the retraction mechanism was not implemented because the simulations in question will not permit "undo" actions. Declaration and Object Management services, which are our main concern, are available according to the version 1.1 of the HLA Interface Specification.

Although the SRTP and DMMC were designed in order to work with DIS, their features can be used to implement the Declaration Management and Data Distribution Management. Under the HLA, the DMMC Application Gateway is easy to convert into a component of the RTI. It becomes a local process, providing RTI services for all the federates on the LAN and routing the data through the federation according to the DDM and the DMMC scheme. We use multicast groups to implement the "stream" abstraction, because when used with SRTP, they provide more appropriate communication services than TCP.

The LWRTI routing space is divided into fixed and equal sized cells (the effectively used extents on each dimension are used here), and each cell is associated statically to a different multicast group. Whenever an HLA region is created, it is associated with overlapping cells and with the corresponding multicast groups. If the region is used as a subscription region, the federate will subscribe to the corresponding groups. If the region is used as an update region, the federate will send the updated attribute values to the mentioned groups.

We implemented the simpler DM filtering using the relevance filtering mechanisms of SRTP. Our LWRTI also is able to support the complex filtering styles of DDM, however greater overhead in processing is incurred. SRTP, with minimal adaptation, can serve as the basis for the data distribution management and declaration management functions of a light-weight RTI. The transport functions of SRTP can be used to provide three of the four types of transport services described in the RTI interface specification as shown in Table 1. The fourth function, fully reliable transport, may be possible to implement for a limited volume of data in the future, using SRTP Mode 1.

Type of Transport Service	SRTP Transmission Mode
Best Effort	Mode 0
Minimum Rate	Mode 0 with heartbeat
State Consistent	Mode 0/1 data stream
Reliable	not supported

Table 1: Providing HLA Services with SRTP and TCP

The data that is being distributed among the federates in an HLA federation can be broken down into attributes and interactions. Attributes are properties of objects, while interactions are events that involve two or more

objects. SRTP, originally designed for DIS, supports the transmission of entity state information (object attributes in HLA terminology) through a data stream comprised of mode 0 and mode 1 data. This carries over into our LWRTI, where we use SRTP modes 0 and 1 to transmit object attribute information. The RTI specification does not make a distinction between a one-to-one interaction (such as a collision) and a one-to-many interaction, such as a radio emission. Therefore the LWRTI does not make use of SRTP mode 2, which was designed to support interactions between exactly two entities, one of which initiated the interaction. The LWRTI transmits interactions using SRTP mode 1, followed by a sufficient number of empty mode 0 transmissions to give a high probability that lost transmissions will be detected, using a multicast transmission that, while it may not be necessary for the distributed simulation, is required by the HLA.

The LWRTI allows two types of filtering, class based filtering and data based filtering. The object and interaction classes defined in the federation object model (FOM) for a federation are mapped to SRTP categories. The SRTP category filtering mechanism is then used to provide RTI class based filtering as part of its declaration management services. As part of the data distribution management scheme, geographic regions are associated with multicast groups. The SRTP multicast group filtering mechanism is used to provide RTI data based filtering. Table 2 describes how the SRTP mechanisms map to HLA concepts.

HLA/RTI Concept	SRTP Mechanism
object/interaction class	category
region	multicast group
Object ID	entity ID
transportation/order type	mode

Table 2: How HLA is Supported by SRTP

2.5 LWRTI Architecture

The LWRTI has a distributed architecture. The different components pass service description information among themselves using TCP to send and receive the services information (which must be delivered reliably and are rather infrequent) and using SRTP to send and receive simulation data, as shown in Figure 2. Each LAN has a Federation Server (an evolution of the DMMC AG) which manages the creation and destruction of federation executions. These servers are fully interconnected using a single multicast group, according to the DMMC concept.

In order to optimize communication, the server should run on the same host as the RTI Gateway. A gateway is not allowed to join the federation after the execution has started (this would cause consistency problems), but may leave if it is no longer supporting any federate on its LAN. In the initial LWRTI design, federates on the LAN are connected to each other and to the RTI Gateway using TCP. (We initially believed that restricting this undesirable non-multicast many-to-many interconnection to the scope of the LAN would provide acceptable performance.)

The RTI Gateway performs the filtering/forwarding tasks according to the DMMC approach and provides RTI services to the federates on the LAN. It also provides the other gateways with aggregated updates for objects being simulated on its LAN. This hierarchical organization simplifies operation of the RTI. A gateway receives subscription and publication from local federates and from remote gateways, and performs data distribution among them. It is aware of the local and remote registered interests in term of classes and regions, and of the locally and remotely simulated objects. It also maps regions to multicast groups for each routing space, whenever a new region is created or when an existing region is modified. Associated with the mapping function is the filtering/forwarding of simulation data. Data sent on the LAN by the federates are forwarded on the WAN using the exercise multicast group if a remote gateway has subscribed to the category of the message, and a remote gateway has subscribed to the multicast group that this message is to be sent on. When the gateway receives a message from the WAN, it checks whether one of the local federates have subscribed to the category and the multicast group; if not, the message is discarded.

3. Performance Issues For The LWRTI Gateway

We developed a very simple federate to test the basic abilities and the correct behavior of the software: objects moving in a two dimensional routing space come in-scope and go out-of-scope and they can send interactions when an other object comes into sight. We also used this federate to determine the performance of the RTI, which was under 100 attribute updates per second. However, testing with a simple federate was not sufficient to validate fully our approach and implementation.

In order to make more appropriate tests at a larger scale, we also used ModSAF 3.0 along with a DIS to HLA translator, driven by DIS Entity State PDUs. The main difficulty in mapping this PDU to the

UpdateAttributeValue service is that the RTI is attribute-oriented, providing only the values of modified attributes, while DIS applications send full-state PDUs at a minimum rate. Therefore, the translator must keep a copy of all known HLA objects and DIS entities. Also, when it subscribes or publishes an object class, it must do so for all attributes of the class, and not only to a subset of the available attributes. If no attribute update has been received recently for a given object, the translator will dead reckon and send an ES-PDU after a given period of time to generate a heart beat.

We conducted a performance analysis of the lightweight RTI, based on several benchmarks and stress tests conducted in the Network and Simulation Lab in the GMU C³I Center. Although no pre-set goals were established on the required level of performance, the overall objective was to maximize the RTI performance within the current design parameters.

The general result of our performance analysis was that communication procedures consume the major of LWRTI processing time and therefore should be further optimized wherever possible. The largest problem which came to light was a communications bottleneck in the Federation Server, which manages the creation and destruction of federation executions. These servers are fully interconnected. In the initial LWRTI implementation federates on the LAN are only connected to each other and to the RTI Gateway using TCP, on the expectation that the relatively higher LAN bandwidth would keep TCP from becoming a bottleneck. Therefore any simulation data such as attribute updates, which transit within the LAN are sent via TCP and not via SRTP. Unfortunately this decision had a considerable negative impact in our test environment where most traffic processed is within the local LAN. Further study showed that tuning the TCP by using a transaction-based API could increase the performance, but the basic bottleneck still remains. An evident solution to the problem is to convert LAN transmission also to SRTP. Even without a change to SRTP, we were able to achieve an increase from 70 to 120 attribute updates per second by locating the test driver on a separate host from the RTI Gateway.

Another problem associated with use of TCP is a considerable per-transaction processing. This problem could be overcome when using SRTP by batching transmissions and using timer to ensure that maximum acceptable latency is sustained. Bilateral preferences can be established between Federates on the latency requirements of real-time updates. This scheme can be implemented within SRTP, where the data can be

buffered and concatenated with other messages going to the same destination.

4. Performance Simulation Of SRTP With HHML

The selectively reliable approach to multicast is based on the clear benefit that can be obtained when data that changes rarely can be transmitted once, reliably, rather than many times, best-effort. As described in [3] and [7], a major drawback in achieving this benefit is the need to achieve reliable multicast transmission for even a subset of all data passing through the LWRTI, because of the congestion that results from proliferation of NAKs when a packet is lost. Accordingly [2] set forth a multi-faceted approach to congestion avoidance, which we seek to implement in SRTP. These included relatively well-understood techniques for NAK suppression and adaptive random backoff to minimize NAK proliferation, which are straightforward in their benefits and implementation. They also included a refinement of Log-Based Receiver-Reliable Multicast [13] to use a multi-level hierarchy of loggers. Although developed independently, our concept is similar to that of Paul [14] in that a hierarchy of loggers is used. However, SRTP is intended for a large-scale distributed simulation environment where senders and receivers may change frequently, making static configuration impractical. Therefore we proposed a dynamic, self-configuring hierarchy in place of Paul’s static hierarchy.

Our approach, called Hop-Hierarchical Multicast Logging (HHML), dynamically generates a hierarchy of loggers in such a way as to localize NAKs for lost packets and resulting replacement messages. The hierarchy is based on the functioning of the underlying IP multicast network. It relies on the feature of IP multicast called packet time to live (TTL) which places a strict limit on the number of hops a multicast packet survives. TTL can be used to localize NAK and replacement activity. Because each router decrements TTL by one when forwarding a multicast packet, TTL can also be used to determine how many hops a packet has come through the network. The general approach in HHML uses:

- loggers at every LAN for every multicast group active on any LAN host
- a hierarchy of LAN loggers that are responsible for circles of increasing radius along the path of the multicast tree for their group, with the first host to join the multicast group serving as the top-level in the hierarchy (see Figure 3)

- a discovery process where hosts newly joining a multicast group determine whether they must serve as logger, and receive initial states consisting of the most recent mode 1 segments that have been sent to the multicast group
- robust fail-safe procedures to ensure that a logger’s place is taken if it dies, and also operate even in the presence of errors in logger discovery and NAK

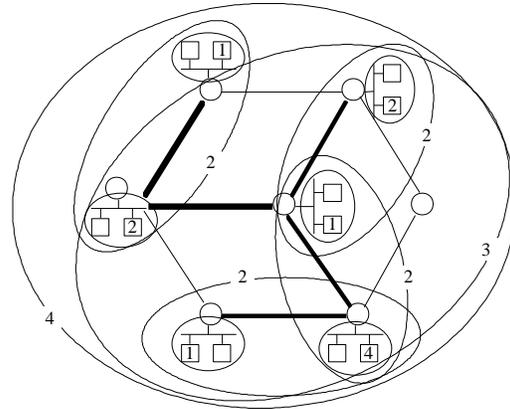


Figure 3: Hop-Hierarchical Multicast Logging (numbered hosts are loggers; dark links represent multicast tree)

Our basic approach is for every SRTP host to perform logger discovery on each new multicast group it joins. If its LAN has no other logger for the group, it becomes the logger. It then probes by means of multicast packets with increasing TTL, and assumes the role of logger for circles of any TTL radius that does not have an established logger. Repeated for every joining host, this will result in full coverage of all hosts while NAKs and repairs will be localized to only those circles of hosts that have lost data.

A problem with dynamic selection of loggers is that the resulting circles may cover some LANs more than once (as for example the rightmost two level-2 circles in Figure 3). We have therefore sought to determine the likely level of overlap (and resulting inefficiency) over a large number of cases. The problem is highly sensitive both to network topology and to the order of entry of hosts to the multicast group, which determines the structure of the multicast tree and the position of loggers at different levels in the hierarchy. In our investigation of the multi-logger-coverage problem, we have used the network shown in Figure 4, which we believe to be representative of the type of network topology likely to be seen in large distributed simulations. Thus far we have confined our investigations to this single, rather complex network topology, with repeated simulations undertaken

to determine the range of results likely to come from different orders of host entry.

A study conducted as part of a graduate networking course at GMU [15] used the Network Workbench simulation system described in [16] to simulate ten cases with different starting hosts. The results are shown in Table 3, which presents the average, best (lowest) case, and worst (highest) case of the ten trials for each logger level (TTL radius).

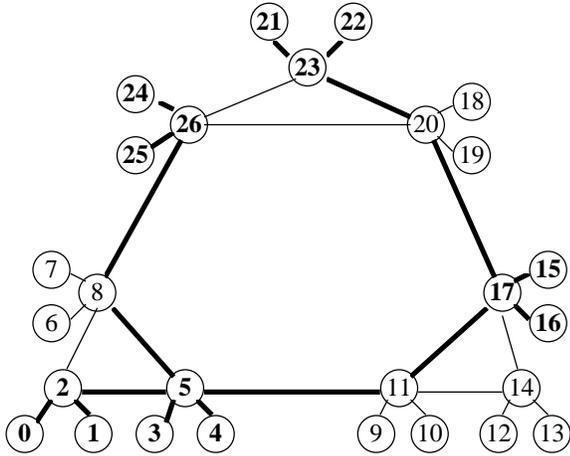


Figure 4: HHML test network
(heavy lines are multicast tree; boldface nodes use SRTP/HHML)

Logger Level	Overall Average	Lowest Average	Highest Average
1	1.48	1.28	1.61
2	2.91	1.72	3.83
3	2.90	1.5	4.5
4	3.17	1.11	5.67
5	2.55	1.33	5.22
6	2.61	1	5
7	1.87	1	2
8	1.80	1	2

Table 3: Level of redundancy in HHML logger coverage for ten random starting hosts

The numbers shown in table 3 indicate that the dynamic logger assignment of SRTP-HHML is reasonably effective in that even under the worst choice of starting host the redundancy only a few duplicate NAKs will be generated. We expect this result will not be sensitive to the network size. It will however remain sensitive to network topology, with long, thin networks having the poorest worst-case performance because they have the longest multicast trees. We intend to investigate

alternative dynamic logger assignment algorithms that may prove to be more efficient.

5. Conclusions And Future Work

We have reported here the results of our continuing investigation of an appropriate implementation for performance of the LWRTI. These results have given us considerable insight into the factors that influence performance under the HLA. We intend to upgrade the LWRTI (1) to use SRTP on the LAN and (2) to use an HHML-enabled SRTP. We believe the resulting LWRTI will demonstrate the considerable utility of SRTP as an RTI component for the class of virtual distributed simulations.

References

- [1] Pullen, J., V. Laviano and M. Moreau, "Creating A Light-Weight RTI As An Evolution Of Dual-Mode Multicast Using Selectively Reliable Transmission", Fall 1997 Simulation Interoperability Workshop paper 97F-SIW-149, September 1997
- [2] Pullen, J. and V. Laviano, "Adding Congestion Control to the Selectively Reliable Transmission Protocol for Large-Scale Distributed Simulation", Fall 1997 Simulation Interoperability Workshop paper 97F-SIW-018, September 1997
- [3] Pullen, J., "The IETF, Reliable Multicast, and Distributed Simulation", Spring 1998 Simulation Interoperability Workshop paper 98S-SIW-208, March 1998
- [4] Cohen, D., "NG-DIS-PDU: The Next Generation of DIS-PDU (IEEE-P1278)" 10th Workshop on Standards for the Interoperability of Distributed Simulations, March 1994
- [5] Cohen, D., "Back to Basics", 11th Workshop on Standards for the Interoperability of Distributed Simulations, September 1994
- [6] Pullen, J. and V. Laviano, "A Selectively Reliable Transport Protocol For Distributed Interactive Simulation", 13th Workshop on Standards for the Interoperability of Distributed Simulations, paper 95-13-102, September 1995
- [7] Pullen, J., M. Myjak and C. Bouwens, "Limitations of The Internet Protocol Suite for Distributed Simulation in the Large Multicast Environment",

Simulation Interoperability Workshop, Orlando Florida, March 1997

Performance Evaluation, George Mason University, Fairfax, VA, June 1998

- [8] Pullen, J. and E. White, "Dual Mode Multicast for DIS", 12th Workshop on Standards for the Interoperability of Distributed Simulations, March 1995
- [9] Frosch, K. and J. Pullen, "Design and Prototype of a Dual Mode Multicast Application Gateway", 1997 Spring Simulation Interoperability Workshop, paper 97S-SIW-121, March 1997
- [10] Defense Modeling and Simulation Office, *DoD Modeling and Simulation Master Plan*, U.S. Department of Defense, October 1995
- [11] Defense Modeling and Simulation Office, *High Level Architecture Interface Specification Version 1.1*, U.S. Department of Defense, February 1997
- [12] Defense Modeling and Simulation Office, *HLA Data Distribution Management: Design Document Version 0.7*, U.S. Department of Defense, November 1997
- [13] Holbrook, H. W., S. K. Singhal and D. R. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation", *Proceedings of ACM SIGCOMM '95*, August 1995
- [14] Paul, S., K. Sabnani, J. Lin, and S. Battacharyya, "Reliable Multicast Transport Protocol (RMTP)", *IEEE Journal on Selected Areas in Communications*, April 1997
- [15] Chang, W. and S. Lee, "Simulation of SRTP with HHML", term project for CS756, Network

- [16] Pullen, J. and V. Laviano, "Prototyping the Selectively Reliable Transport Protocol", 14th Workshop on Standards for the Interoperability of Distributed Simulation, paper 96-14-089, March 1996

Author Biographies

J. MARK PULLEN is an Associate Professor of Computer Science and a member of the C³I Center at George Mason University, where he heads the Networking and Simulation Laboratory. 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 and a Fellow of the IEEE. Dr. Pullen teaches courses in computer networking, and has active research in networking for distributed virtual simulation and networked multimedia tools for distance education. Dr. Pullen received the IEEE's Harry Diamond Memorial Award in 1995 for his work in networking for distributed simulation.

NAGESH KAKARLAMUDI is a doctoral student in Information Technology at George Mason University. He holds the BSEE degree from BITS, India and the MSIE degree from NITIE, India. He is employed with Performance Engineering Corporation in Fairfax, Virginia, where he is involved in numerous projects involving computer system performance and information security. He is a Member of the IEEE.