NGI-FOM over RTI-NG and SMRP: Lessons Learned

J. Mark Pullen, Robert Simon, Fei Zhao and Woan Sun Chang
Department of Computer Science and C3I Center
George Mason University, Fairfax VA 22030
{mpullen, fzhao}@netlab.gmu.edu
{simon, wchang}@cs.gmu.edu

Keywords:
reliable multicast, High-Level Architecture, runtime infrastructure

ABSTRACT: *George Mason University has developed the Selectively Reliable Multicast Protocol (SRMP) with support from the Defense Modeling and Simulation Office. SRMP allows reduced network traffic in distributed virtual simulation by sending rarely changing data reliably, frequently changing data best-effort, and exploiting the synergism between the two to provide for reliable delivery. We have integrated SRMP with the RTI-NG and used it to support a Next-Generation Internet FOM, consisting of open-source distributed virtual simulation software from the Naval Postgraduate School, adapted with a gateway developed by GMU. The software we have developed is available open-source.This paper summarizes the benefits we have been able to achieve and the problems we solved in achieving them.*

1. Introduction

In the past several years, the Networking and Simulation Laboratory of the C3I Center at George Mason University (GMU) has been investigating an approach to reliable multicast for virtual simulation (DVS) that exploits the synergy between reliable and best-effort transmission supporting DVS over multicasting networks. This paper describes lessons learned in the process of integrating our Selectively Reliable Multicast Protocol (SRMP) with the RTI-NG to support an experimental Federation Object Model for use over the Next Generation Internet, the NGI-FOM.

2. Selectively Reliable Multicast

There is no general approach to achieving a reliable multicast (RM) protocol that achieves reliable, ordered delivery as the Transmission Control Protocol (TCP) does for unicast. All approaches to RM take advantage of some property of their application domain. Our Selectively Reliable Multicast transport Protocol (SRMP) supports a mix of reliable and best-effort multicast by taking advantage of the specific requirements of DVS [1]:

- Most traffic consists of data streams describing position, orientation, etc., that are refreshed frequently and thus may be distributed by best-effort transport.

- A small fraction of the traffic consists of data that change rarely and thus require reliable transport.

- Generally, it is straightforward to determine which data elements fall in each category, within a higher-layer protocol such as IEEE 1278 Distributed Interactive Simulation (DIS) or a Federation Object Model (FOM) based on the IEEE 1516 High Level Architecture (HLA) [2].

- Only the latest value of any data element is required to be delivered reliably; therefore, during times of rapid transition, it is not essential to deliver all reliable data element changes in sequence.

Following a concept originated in [3], SRMP originally was developed for use with Distributed Interactive Simulation (DIS). [4] reported on a “Light Weight RTI” that used SRMP to support a subset of HLA functionality. [1] demonstrated that the functionality of SRMP can be applied to good effect in the HLA if the reliable transport function is separated from other functionality of the Run Time Infrastructure (RTI). The performance advantages of SRMP under these circumstances, as projected by simulation, have been described in [5]. Recently, we have implemented SRMP in a running HLA federation and have shown that many of our analytical and simulated projections are approximately correct. [6] and [7] report on the construction of SRMP and how it fits in the NGI FOM. [7] and [8] report on our design for this project. The next section provides a synopsis of this design and our experience in implementing it.

3. Using SRMP with the RTI-NG

Standardized HLA applications and implementations use the Run-time Infrastructure (RTI) as the middleware glue between communicating federations. Among its functions, the RTI provides distributed object and communication management at runtime for all participating federations. RTI implementations also provide programmers the interface libraries for developing distributed simulations.

Most work in RTI software has used existing transport level protocols TCP and UDP for communication among the distributed components of the RTI. However, these protocols cannot provide the efficiency benefits of SRMP. It was therefore necessary to extend the capabilities of the RTI to support a mixture of communication styles. We have done this by extending the facilities used by the RTI-NG to access the underlying network. Specifically, we have provided for the RTI-NG an intelligent *Event Channel* service, capable of supporting the semantics of the NGI FOM and taking advantage of SRMP. The event channel manages and communicates with the SRMP daemon, and provides NGI FOM semantics for HLA applications.

3.1 Event Channels in the RTI

A common approach in building large-scale multi-party distributed systems is to connect data producers with data consumers via a communication paradigm known as an *Event Service*. The Event Service is defined in the OMG Common Object Services Specification (see http://www.omg.org) for the Common Object Request Broker Architecture (CORBA) [10]. This service definition is useful because it decouples producers and consumers, and therefore eliminates the requirements that these entities have explicit information about each other. The basic programming model for using an Event Service is the Event Channel. Each Event Channel appears as a proxy or mediator between one or more producers and one or more consumers. (In standard CORBA documentation, producers are called "suppliers.") Both pull and push techniques can then be applied to the channel. Producers can push data to consumers, and consumers can pull data from producers. This model is very useful for group communications, since multiple producers and consumers can connect via a single channel. Further, Event Channels are specified at the data object level, so they are particularly useful to support HLA applications. In addition to the push and pull models, event channels can be used in a hybrid mode. One such mode is to make the event channel an intelligent agent, so that the channel itself makes push and pull decisions. An advantage of this approach is that user-configurable policies can be implemented for scheduling and data transmission.

We developed an SRMP-capable event channel and integrated it with the DMSO RTI-NG and its supporting middleware, TAO. Our FOM is based on the Naval Postgraduate School's open-source implementation of DIS with Java and VRML (available via http://www.web3d.org/WorkingGroups/vrtp/), plus a DIS-HLA gateway (described in [8], available open-source at http://netlab.gmu.edu/SRMP) based on the RPR-FOM [9]. Figure 1 shows the relationship of the HLA, the RTI-NG, TAO, our event channel, and SRMP.

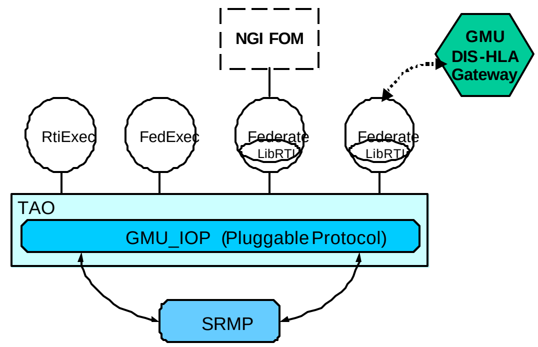

Figure 1: GMU enhancements to the TAO/RTI runtime system

The components in Figure 1 all have service configuration files used at runtime to configure services, including specifications associated with the NGI FOM and the *GMUGateway*. As part of our project we have developed a general-purpose configuration language based upon XML. The most useful attributes of XML for this application are that it is human readable and self-identifying. The purpose of the configuration language is to specify the run-time characteristics of the RTI, in order to encapsulate the standard RTI and FOM initialization files (.fed and .rid), and to explicitly provide SRMP configuration data including the list of multicast addresses. We have implemented a thin wrapper function within our RTI that parses this configuration file in order to comply with the standard .fed and .rid format. Based upon the configuration file, when the RTI is invoked, the event service module dynamically provides multicast group information to initialize SRMP. After an Event Channel is initialized, reliable data identifiers are associated with this information using SRMP's registration service. Other primitives used for Event Channel Management include subscription, sending data, and receiving data.

Our development could not begin until we had an effective grasp on the internal structure of TAO and the ways in which the RTI-NG uses TAO. This proved quite complex; TAO is well documented but its internal structure is quite arcane. RTI-NG in turn consists of tens of thousands of lines of well-commented code but had only minimal system-level documentation available when we undertook this analysis. After significant effort, we determined that the internal relationship between RTI-NG and TAO is quite complex, as shown in Figure 2.

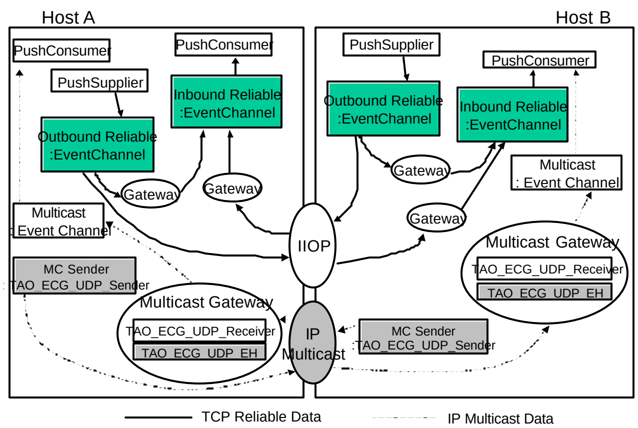

Figure 2: RTI-NG and TAO Communications

3.2 SRMP used with the RTI-NG and TAO

Figure 3 shows a portion of our Event Channel model. The diagram depicts the use of two types of event channel services. Unicast reliable data is sent to event channels set to Push mode. The purpose of the gateway (which should not be confused with our HLA-DIS gateway, described above) is to multiplex and demultiplex packets from the GMU-IOP to the appropriate event channel. Notice that it is possible to have both consumers and suppliers operate in the same host. This requires the use of a gateway within the host. This portion of the design is consistent with the current DSMO RTI release. We have added a multicast event channel for intelligent processing, as the interface between the current multicast send primitive available in the RTI. Multicast applications PUSH data to this event channel. On the receiver side, the multicast gateway demultiplexes packets from the GMU-IOP and delivers them to the appropriate event channel. The version of RTI-NG that we worked with does not use an event channel for multicast data; therefore, we added this capability, on the receive side only, in order to enable use of SRMP.

An unanticipated problem we encountered was supporting setup of the RTI-NG at the beginning of federation execution. As described earlier, SRMP does not guarantee ordered delivery of reliable data, only reliable delivery of the last transmission, and has the property that it cannot deliver transmissions reliably without a co-existing best-effort stream. The RTI-NG requires ordered delivery during setup, so we left its TCP connections in place for setup and used SRMP to deliver the simulation data after setup. SRMP also requires the application to identify specific data elements for reliable transmission. The required information is documented in the FOM by way of the .rid file, so we created a mapping between the RTI

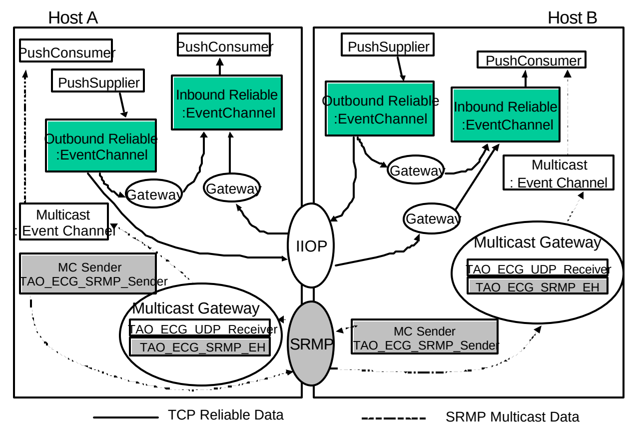

Figure 3: GMU adaptation of the TAO/RTI-NG runtime system,with SRMP

transportModeID (ObjectClassHandle:AttributeHandle: InteractionHandle) and the DataID codes used to identify specific reliable elements for SRMP. However, the RTI-NG accesses the transport layer via TAO middleware and thus has no direct path to identify reliable and best-effort transmissions to SRMP. With the assistance of the RTI-NG's developers, we were able to identify an unused data element that passes through the RTI to TAO and from there to other RTI elements with every data transmission, and use this to pass the DataID code to SRMP. The partitioning of data into reliable and best-effort is shown in Table 1. To keep the best-effort data synchronized, we combined all best-effort attributes into a single meta-attribute for transmission. Similarly, the reliable attributes were combined into a single aggregated attribute in order to minimize header overhead due to the DataIDs that SRMP transmits with each bundle of HLA messages.

Best-Effort	Reliable	Reliable
VelocityVector	BaseEntityClass	PowerPlantOn
Position	EntityID	RampDeployed
Orientation	Entitytype	SmokePlumePresent
IsFrozen	DRAlgorithm	TentDeployed
AccelerationVector	DamageState	TrailState
AngularVelocityVector	EngineSmokeOn	Marking
	FlamesPresent	HasFuelSupplyCap
	HatchState	HasRecoveryCap
	LifeformState	HasRepairCap
	LightsState	ArticulatedParameters Array

Table 1. Reliable and Best-Effort Attributes in NGI-FOM

We demonstrated SRMP using the NGI-FOM with ten federates at GMU, NPS, and Old Dominion University (ODU) over the Next-Generation Internet. The demonstration architecture is shown in Figure 4.

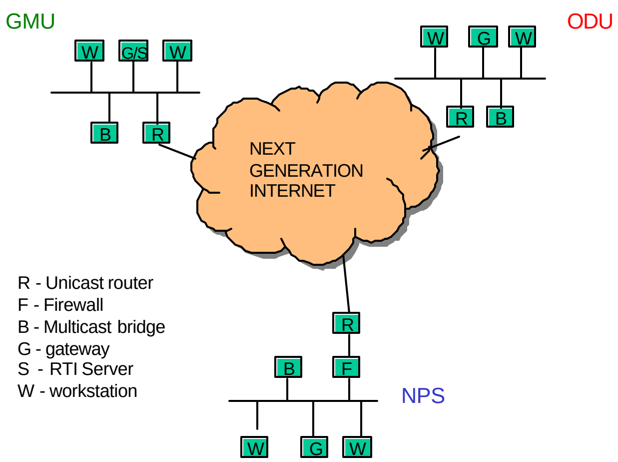

Figure 4. SRMP demonstration over NGI

4. SRMP Performance Predictions and Measured Results

We have collected some interesting results that show the potential value of SRMP in the HLA environment.

4.1 SRMP gain over TCP

Because one mode of the RTI-NG's operation uses TCP to achieve reliable transmission, we could compare the network requirements of SRMP versus TCP. We emphasize here that there is no magic in SRMP. It takes advantage of known characteristics of DVS, in that a portion of the data to be transmitted represents constantly changing state that cannot benefit from reliable transmission because any repair to a network loss would arrive after an intervening update and therefore be useless. At the same time, this stream of best-effort data can be used to piggy-back codes that indicate when a reliable transmission has been lost. Combined with a standard technique to support negative acknowledgements (NACKs) as described in [4], this allows the rarely-changing data to be sent reliably only when it changes. Both sets of data are sent by multicast, resulting in a reduction of network traffic proportional to the number of federates relative to TCP, as shown in Figure 5. The figure assumes 122-byte payload; negligible loss rate; no bundling; one reliable DataID. (The 122 bytes represent a 32-byte RTI header, 58-byte TAO header, and 32 bytes of application data.) Again, there is no magic; with TCP, each federate sends a separate copy of each attribute to every other federate. SRMP sends a single copy to all federates via multicast. Figure 5 represents data measured during our demonstration. Although it has roughly the expected linear shape, it falls short of the expected gain, which would be one less that the number of federates because the overhead with TCP and SRMP is almost identical. We suspect that TCP backoff due to the round-trip time of the WAN had the effect of reducing overall TCP transmission and therefore masking part of the gain. However, we did not collect enough data to confirm this.

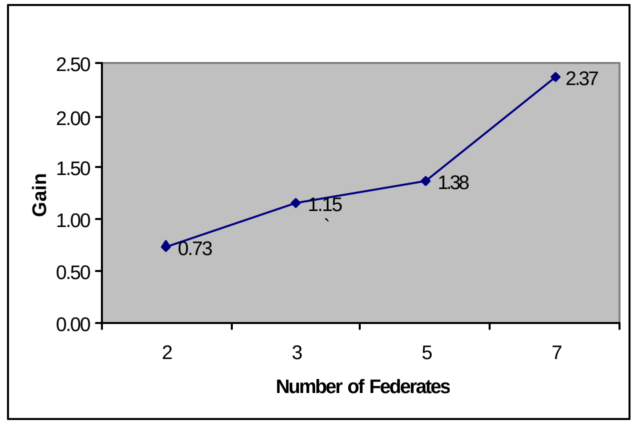

Figure 5. Measured SRMP gain over TCP

4.2 SRMP efficiency

In order to minimize protocol overhead, SRMP bundles transmissions that are presented by the application within a configurable *bundle timeout* period.

Actual efficiency of SRMP depends on several factors:

- Underlying network packet loss rate, which causes retransmissions; in a modern network that is not overloaded, this should be minimal, as overloading that would result in heavy packet loss due to congestion would also make the network unsuitable for real-time simulation
- Number of reliable DataIDs, which increases overhead because it requires larger bundle headers.
- Data message length; as this increases, the number of messages per bundle is reduced, resulting in a lower ratio of useful data to overhead
- Ratio of data rate to acceptable bundle timeout; as this falls, more bundles must be sent incomplete. However, if the simulation is presenting a low data rate it typically is not stressing network capacity, so lower efficiency may not represent a problem.

Thus, the dominant factors in efficiency are number of messages in the bundle, induced by data message length, and number of reliable DataIDs. Figure 6 shows the relationship between these quantities. For number of DataIDs less than 64, SRMP efficiency is generally above

70% and always above 50%. Combined with observed gain, this implies that SRMP will be better than TCP for any number of federates over three, for up to 63 DataIDs.

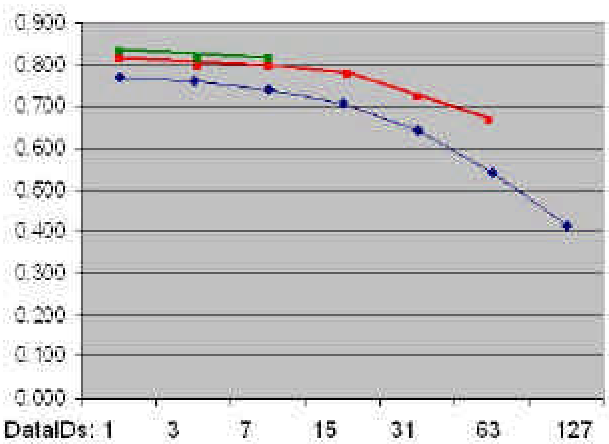

Figure 6. SRMP Efficiency

4.3 SRMP and TCP-friendliness

We have introduced SRMP into the Internet standards process, which is conducted by the Internet Engineering Task Force (IETF), an association of mostly industry technologists. The IETF is known for an approach based on "loose consensus and running code," however its executive body, the Internet Engineering Steering Group (IESG) works aggressively (and with evident success) for stability, performance, and service in the Internet. When the IETF Reliable Multicast Transport (RMT) working group was created in 2001 to standardize building blocks for bulk data transfer, the IESG stipulated that any multicast transport protocol standardized for Internet use must be designed to compete fairly with TCP, in the sense that it reduces transmission in roughly the same way TCP does whenever network congestion is encountered. The various protocols put forward by RMT have all converged on a building block called TCP-Friendly Multicast Congestion Control (TFMCC) developed by Widmer and Handley [11].

Our analysis indicated that it would be possible to extend TFMCC's one-to-many congestion control into the many-to-many paradigm for DVS. As part of our work with SRMP, we have designed such an extension and added it to our SRMP implementation. Where the RMT protocols for bulk data transfer reduce network load by slowing the rate of transfer, we perform the reduction by random drop of best-effort data, which by definition can tolerate some losses without seriously degrading the simulation. Initial results indicate that the result does in fact behave like TCP, and also that it does support DVS with dead reckoning, albeit in a degraded form, at packet loss rates up to 30%. More experimentation will be needed to determine if this is a practical approach that exhibits satisfactory TCP-friendliness.

5. Conclusions and Future Work

A summary of our results is:

- We have demonstrated that it is possible to use SRMP in an RTI to transmit real-time simulation data by a mix of reliable and best-effort multicast, resulting in reduction of network traffic by a factor proportional to the number of federates, with net improvement beginning at three federates.
- Our experience with the RTI-NG and TAO indicates that retrofitting a new transport protocol into highly complex middleware is time-consuming and is best undertaken by the middleware developers, or at least with the close developer cooperation we had. It also seems likely that the process would be simpler for an RTI with less complex underlying software than TAO.
- We have further completed an initial demonstration that TCP-friendliness is possible in SRMP, as required for Internet standardization.

We began this project expecting that the NGI FOM and the software associated with it would be the starting point of a new type of involvement by the academic research community in large-scale virtual environments for distributed virtual simulation. This is still likely to be true, however in a somewhat different sense than we expected. In the intervening eighteen months, DMSO has discontinued development and distribution of the RTI-NG in favor of commercial RTI production. Thus, while our demonstration of the merits of SRMP remains valid, the idea that the RTI-NG (with or without SRMP) will serve as experimental infrastructure has lost its appeal. At the same time, GMU has been working with NPS and others to demonstrate the merits of Web-based simulation and the results of this work have clear applicability in the context of the Extensible Modeling and Simulation Framework (XMSF) [12].

We see three areas where more work is called for in the domain of SRMP:

- Considerably more experimentation with TFMCC and its likely successors will be needed in order to arrive at a stable protocol that can support DVS under a range of network loading conditions while competing fairly with TCP.
- SRMP itself needs further development to take advantage of work recently completed in the IETF-RMT on tree-based repairs and forward error correction. Either or both of these could provide a basis for ordered, reliable delivery, a service for

which there is growing demand as C4I systems are coming together with DVS and sharing the same networks.

- There is growing interest in overlay multicast techniques, particularly end system multicast, which can provide a network service like IP multicast, with performance that approaches that of multicast packet switching, over parts of the Internet where IP multicast is not implemented (and is not likely to be, due to the business model of the service providers). SRMP with TFMCC should be combined with end system multicast to achieve the benefits of both approaches.

Acknowledgements

This work was supported in part by the Defense Modeling and Simulation Office High Level Architecture RTI Enhancement Program. We would also like to acknowledge assistance from Roger Wuerfel of SAIC Inc., the developer of the RTI-NG, in arriving at the software solutions described in section 3.

References

[1] Pullen, J. M., "Reliable Multicast Network Transport for Distributed Virtual Simulation," *Proceedings of the Third International Workshop on Distributed Interactive Simulation and Real Time Applications*, IEEE, 1999

[2] US Department of Defense Modeling and Simulation Office, "High Level Architecture Run-Time Infrastructure RTI 1.3-Next Generation Programmer's Guide", 1999

[3] Cohen, D., "Back to Basics", 11th Workshop on Standards for Interoperability of Distributed Simulations, September 1994.

[4] Pullen, J.M., V. Laviano and M. Moreau, "Creating A Light-Weight RTI As An Evolution Of Dual-Mode Multicast Using Selectively Reliable Transmission," Simulation Interoperability Workshop, Orlando, FL, September 1997

[5] Moen, D. and Pullen, J.M., "A Performance Measurement Approach for the Selectively Reliable Multicast Protocol for Distributed Simulation," *Proceedings of the IEEE Distributed Simulation and Real Time Applications Workshop(DS-RT'01)*, Cincinnati, OH, August 2001

[6] Shanmugan, B., and Pullen, J.M., 'Software Design for Implementation of the Selectively Reliable Multicast Protocol," *Proceedings of the IEEE Distributed Simulation and Real Time Applications Workshop (DS-RT'02)*, Dallas, TX, October 2002

[7] Pullen, J.M., Simon, R., Khunboa, C., Parupalli, M., and Brutzman, D., A Next-Generation Internet Federation Object Model for the HLA, *Proceedings of the IEEE Distributed Simulation and Real Time Applications Workshop (DS-RT'02)*, Dallas, TX, October 2002

[8] Pullen, J. M., and R. Simon, "Selectively Reliable Multicast for the HLA," Simulation Interoperability Workshop, Orlando, FL, September 2002

[9] Berkeley, J., "Enhancing The RPR FOM To Include STOW 97 Capabilities," Simulation Interoperability Workshop, Orlando FL, September 1998

[10] Common Object Request Broker Architecture. Object Management Group, July 1995

[11] Widmer, J. and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification," work in progress, Internet Engineering Task Force, November 2002

[12] 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

Author Biographies

J. MARK PULLEN is Professor of Computer Science at George Mason University, where he heads the Networking and Simulation Laboratory in the C^{3}I 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.

ROBERT SIMON is Associate Professor of Computer Science at George Mason University. He holds the M.S. in Computer Science from New York University and the Ph.D. in Computer Science from the University of Pittsburgh. He is active in teaching and research in network and distributed system architectures, protocols performance analysis and simulation methods for real-time, multimedia and computer supported cooperative work applications, and is known for his work in network simulation. He has served as Program Chair for Society for the Computer Simulation Computer Networks and Distributed Systems Conference 1999 2000 and 2001.

FEI ZHAO received the M.S. degree in Electrical Engineering from Tianjin University, China in 1997. He worked as a network engineer and architect in Jitong Communication Co, China between 1997 and 2001. He is currently a PHD student in Computer Science at George Mason University, and working as a Research Assistant in Networking & Simulation Lab. His research interests include network multicasting and traffic control.

WOAN SUN CHANG is a doctoral student at George Mason University. She received the M.S. degree from George Mason University in 1994. In the C3I center, she has worked as a Research Assistant on the integration of SRMP and RTI and developing the gateway for the interoperability of Distributed Interactive Simulation (DIS) and High Level Architecture (HLA). Her research interests are Quality of Service Management for Computer Networks, overlay multicast, and Network Pricing.