As published in *Proceedings of the 2001 IEEE Distributed Interactive Simulation and Real-Time Systems Workshop*

A Performance Measurement Approach for the Selectively Reliable Multicast Protocol for Distributed Simulation

Dennis M. Moen
Primus Telecommunications
McLean, Virginia
dmoen@primustel.com

J. Mark Pullen
Dept. of Computer Science and C3I Center
George Mason University
Fairfax, Virginia
mpullen@gmu.edu

Abstract

The Selectively Reliable Multicast Protocol (SRMP) provides an approach to reliable multicast that is specialized to distributed virtual simulation. SRMP operates in three modes of reliability: best-effort transmission of transient data that does not need reliable transport; reliable transmission of object state data; and timely, reliable transport of data to a single, dynamically determined receiver within the multicast group. This paper deals with the first two modes, which dominate SRMP network performance. Any multicast protocol, when used across wide area networks to share information of among thousands of objects in a real time simulation, will create traffic flows with many complexities. In order to evaluate the merits of using SRMP, a mechanism to measure its performance across a network is necessary. This paper presents an approach to measure the bounds of performance of the SRMP in real time simulation environments. The approach is based on a Markovian model of object activity in the simulated environment, and is implemented in the context of a cluster of workstations that emulate wide-area network performance using our protocol. We present initial results of our SRMP evaluation in this environment.

1. Introduction

The Selectively Reliable Multicast Protocol (SRMP) [1] was developed at George Mason University to support large scale distributed virtual simulation (DVS) [2], initially using the acronym SRTP. A number of protocols have been proposed for reliable multicast transport, for example RMTP [3]. Our observation is that each reliable multicast protocol is applicable in only a restricted application domain [4]. SRMP's application domain is defined by the ability to use selective reliability to support reduction of transmitted state information in DVS using known temporal characteristics of object attributes. The approach is based on ideas developed in Cohen's work with Distributed Interactive Simulation [5,6]. We have extended its use to the High Level Architecture (HLA) for modeling and simulation [7].

The SRMP, used with the User Datagram Protocol (UDP), provides a transport service that embodies a tradeoff between minimal latency and reliability. This is achieved by exploiting the known properties of distributed simulation traffic. Network capacity requirements are reduced by using real-time best-effort multicasting to transmit those data elements that change frequently, such as position. Those data that change rarely, or not at all, are transmitted reliably one time. Furthermore, using SRMP's point-to-point mode, transactions between any two hosts in the multicast group can be achieved reliably without incurring the delay associated with setting up a TCP connection. The SRMP, therefore, provides a service model whereby a group of real time distributed simulators send and receive data, exchanging information about the current state of objects in each simulator group using a well-chosen mix of reliable and best-effort transmissions.

The number of distributed objects in the DVS environment can scale to many thousands. This has the potential to create a significant demand for network capacity in what might already be a congested environment. In addition, group members may be distributed over wide geographic areas, potentially worldwide [8]. Clearly, this is a very complex environment. Consequently, measuring the performance of the SRMP protocol is difficult even in simple configurations. This paper describes our work to create an effective method of evaluating the performance of SRMP.

2. Model Design

In this section we describe how SRMP works, and how we modeled our application to study its performance.

2.1 Characteristics of SRMP

For the SRMP, each of the three modes of reliability has different performance criteria for success:

- Mode 0: Transient data does not need reliable transport, therefore performance is limited by capacity and throughput of the network. Ideally, throughput is the same as the offered load. However, our plans for SRMP do include congestion control achieved by selectively dropping Mode 0 messages at the source. Since errors are tolerated for the transient data of real time distributed simulators, throughput can be somewhat less than the offered load without seriously affecting performance of the distributed simulators.
- Mode 1: Reference data in an object-state protocol requires reliable transport. This data must be received reliably while maintaining low end-to-end delay. SRMP is intended to provide reliable service for this data, while meeting the demands for throughput and low end-to-end delay of the real time simulators. This is accomplished by including information in the Mode 0 messages to indicate when a Mode 1 update has occurred, allowing loss detection leading to generation of a negative acknowledgement (NACK) and subsequent retransmission of the data.
- Mode 2: The third type of data exchange is a transaction with a single, dynamically determined receiver in the multicast group. This also requires reliable and timely transport. This capability was originally developed to meet the DIS requirement for exchange of collision information. It appears likely also to prove useful for federation setup under the HLA. However, in any event it is expected to be used for only a small fraction of all network traffic and thus will have a very small impact on network performance. Accordingly, it will not be considered further in this paper.

Because SRMP is intended for support of real-time multicast traffic, congestion control is extremely important to its performance. Even moderate amounts of congestion can destroy the real-time performance necessary for effective real-time simulation. SRMP has been implemented in an initial form that supports the three modes of functionality and also has very simple congestion control in the form of NACK suppression. We are now beginning research to evaluate other methods of congestion control proposed in [4]: a dynamically constructed tree of loggers for localized repair of losses, bundling of messages, and adaptive dropping of mode 0 messages to reduce network workload.

2.2 Application Model

We have created a model to study the measures of interest for the SRMP including packet loss probability, delay and throughput. An approach using these measures is needed to measure relative performance in an open network environment, including a representation of typical traffic patterns generated by objects in distributed simulation. We are interested in a performance measurement model that will assist in addressing the following questions and help establish the upper performance bounds of SRMP:

1. How scalable is the protocol with regard to the number of users in a multicast group?
2. What are the mechanisms that limit scalability, and how can we estimate their limits?
3. How well does the protocol protect from congestion?
4. Under what circumstances will the protocol fail to perform functions needed by the application?
5. How well does the congestion control mechanism perform, and when does it fail?

2.3 Performance Model

Our approach is to build a statistical traffic generator that generates packets representing the three modes and to use the SRMP to transmit these packets across both a local area network and a wide area network. The concept is presented in the figure 1, Statistical Traffic Generator. We develop a statistical model, the output of which drives a state counter that maintains the state of each simulation objects at all times. Real time messages are generated at a rate scaled to the size of the counter for each state and mode or type of traffic. The model represents an on-off source traffic model that can be described as a discrete event, continuous-time Markov chain.

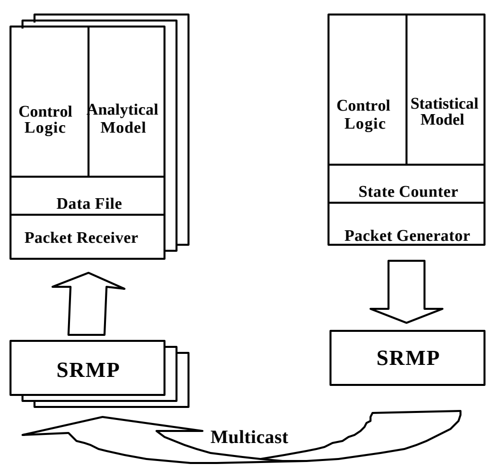

Figure 1. Statistical Traffic Generator

2.4 State Counter

We begin by defining a four-state process as a representation of the status of an object in a distributed real time simulator (figure 2). These four states are:

1. Object *Off* state: In this state the object is "off" and is not generating any packets.
2. Object *Idle* state: The simulation object is active in the visual space of the simulation, but is not moving and is stable in a multicast group. In this state, the object will generate Mode 0 packets at fixed rate. The object will infrequently (exponential distribution) generate Mode 1 packets.
3. Object *Moving* state: The simulation object is moving within the current multicast group. In this state the object will generate Mode 0 packets at a fixed rate. The object will infrequently (exponential distribution) generate Mode 1 packets.
4. Object *Join* state: The simulation object joins a new multicast group. In this state, the object will generate Mode 1 packets at fixed rate until established in the new multicast group.

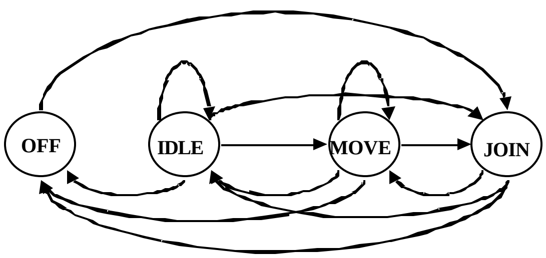

Figure 2. Object State Transition Diagram

This model describes a discrete event continuous-time Markov chain [9]. The state transitions are governed by a Markov chain in which the dependency of the successive state of the simulation object is only dependent on the object's current state. The holding times in each state are, however, exponentially distributed, therefore defining a continuous-time Markov chain.

The traffic generator model is defined as a continuous-time stochastic process $\{X(t), t \geq 0\}$ with discrete state space I and

$$P\{X(t_n)=i_n|X(t_0)=i_0,\ldots,X(t_{n-1})=i_{n-1}\} = P\{X(t_n)=i_n|X(t_{n-1})=i_{n-1}\}$$

for all $0 \leq t_0 < \ldots < t_{n-1} < t_n$ and $i_0, \ldots, i_{n-1}, i_n \in I$.

Under this definition, we establish two rules under which we allow the discrete state space to change state:

1. If the system jumps to state I, it stays in state I an exponentially distributed time with a mean μ independently of how the system reached state I and how long it took to get there.
2. If the system leaves state I, it jumps to state j with probability $p_{ij}(j \neq i)$ independent of the duration of the stay in state i, where $\sum_{j \neq i} p_{ij} = 1$ for all $i \in I$.

Given this definition of the model, a traffic generator is programmed to represent this four-state model with a set of rules representing the activity of an object in a distributed real time simulation. Packets are generated and transmitted to the multicast group at a rate and type (Mode) for each object based on the current state of the object. In the initial characterization of the model, all distributions are assumed to be exponential, though the model is built to allow substitution of different distributions for special case analysis or where well known distributions exist that prove to represent distributed virtual simulation data better than the exponential.

3. Concept

SRMP must perform across large congested networks including the public Internet and large private distributed simulation networks. The model presented above is implemented as an application that can run on multiple SRMP clients over real networks. This allows performance measurement in a live network environment. The traffic generator is intended to be run on a set of servers, initially in a lab environment for baseline performance measurements, then across a wide area network such as the Internet Multicast backBone (MBone). The client will time stamp and sequence number the packet generated and then forward to a multicast group. Each member of the multicast group will receive SRMP packets and collect their arrival statistics. From this data, relative performance statistics can be calculated for throughput, delay and lost packets.

3.1 Wide-Area Network Emulation

We require the ability to evaluate performance of SRMP congestion control mechanisms in a wide-area network environment. However, such an environment is problematic in that real wide-area networks to which we have access (the Abilene network and the Defense Research and Engineering Network, both part of the Next Generation Internet) display constantly changing performance as their usage varies. We might have turned to a network simulator such as OPNET to evaluate SRMP. However, our experience with this approach indicates that we would not be able to study an extended period of simulated time because, for large configurations, hours of simulation time are required to represent minutes of simulated time [10]. In order to have a reproducible testing environment capable of real-time operation, we have adopted emulation of the target system much like that used in NISTNet (see http://www.itl.nist.gov/div892/itg/carson/nistnet/), and ns (see http://www-mash.cs.berkeley.edu/ns/ns.html), but customized to our situation. We represent the latency of the WAN by a configurable delay, with configurable, randomized packet dropping inserted between SRMP and UDP as shown in figure 3. We run the emulation on a system of high-performance workstations connected through a Myrinet gigabit-per-second LAN. This approach lets us exercise the actual SRMP protocol code in a way that enables evaluation of its effectiveness to meet the needs of DVS while reducing overall network requirements.

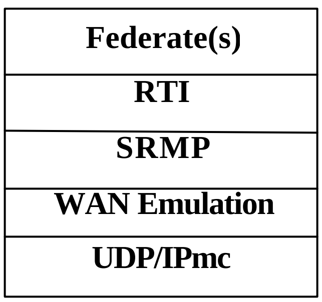

Figure 3. SRMP in Protocol stack with WAN emulation

4. Initial Results

We have implemented a simplified version of the model in figure 2 using the protocol stack in figure 3 on a cluster of four workstations. Each workstation supported fifteen simulated objects using one SRMP daemon with a mix of 99% Mode 0 and 1% Mode 1 data, representing objects with alternating, exponentially distributed idle and moving periods. The results are shown in table 1.

The fact that Mode 1 retransmissions are on the same order as Mode 1 packets transmissions might appear to be a deficiency. However, because we do not yet have a mechanism in SRMP for localizing repairs and NACK suppression is much less effective on the WAN than it would be in a LAN, this level of retransmission is consistent with the fact that every Mode 1 message was delivered to 60 objects. Thus, the imposed loss rate of five percent could potentially have generated over 6,500 retransmissions. The 2,220 actually experienced represents less than one percent of overall network traffic.

Table 1. Experimental Results

Variable	Value
Number of WAN locations	4
Number of objects per location	15
Duration	30 minutes
Packet size	2000 bits
Simulated packet drop	5%
Portion of time objects active	50%
Mean period of activity	50 seconds
Idle state frequency	0.5 packets/sec
Moving state frequency	5 packets/sec
Mode 0 packets transmitted	296,684
Mode 1 packets transmitted	2,169
Mode 1 retransmissions	2,220

We note that these statistics illustrate the potential of SRMP to support distributed simulation with reduced network capacity, in that they support a federation of 60 objects using about 330 kb/s of multicast network capacity. Using the hybrid reliable multicast developed for the Synthetic Theater of War (STOW) [11], we estimate that 12 TCP connections requiring a total network capacity of 1.0 Mb/s at each site would be required. While reducing this to 330 kb/s represents a significant reduction, as we scale SRMP to larger configurations the potential savings are much greater. This is because the reduction in network capacity requirement under SRMP is proportional to the number of WAN nodes. However, considerable more work will be required to produce a robust protocol that scales beyond the current 60 objects and deals effectively with network congestion by methods such as a hierarchy of distributed loggers. We expect the WAN emulation will help us to achieve those goals.

We have been able to validate the statistics produced in our WAN emulation by using an open Jackson network analytical model (M/M/C) [12]. Our results indicate that the above samples are well within expected performance using 384kb/s links, specifically a packet sojourn time of less than .04 seconds with an estimated system size of 4 packets. We also recalculated the model assuming TCP (which implies adding ACK packets for all source packets generated to the network load, assumed to be an order of magnitude increase in the number of packets). In order to obtain similar performance parameters, the network links must now be greater than T1 (1.54Mb/s), which is consistent with our comparison to STOW results.

5. Conclusion

We have developed an approach to performance measure for SRMP using a discrete event, continuous time Markov model as the basis for development of a traffic generator for use in evaluating protocol performance. The approach gives a tool that can be used for understanding SRMP performance in a real network environment. With this approach, we expect to define upper bounds for SRMP performance in a congested network environment. This approach will allow studies and analysis of multicast protocols to address such topics as:

- Scalability: How scalable is the protocol to the number of objects in a multicast group?
- Identification of mechanisms that limit scalability and estimate that limit.
- Congestion: implement and validate mechanisms for congestion control.
- Functionality: Under what circumstances will the protocol fail to perform functions needed by the application?

6. References

[1] Pullen, J. and V. Laviano, "A Selectively Reliable Transport Protocol for Distributed Interactive Simulation," *Proceedings of the 13th Workshop on Standards for the Interoperability of Distributed Simulations, 1995*

[2] Pullen, J., "Networking for Distributed Virtual Simulation," *Computer Networks and ISDN Systems Journal*, Vol. 27, 1994, Elsevier North-Holland

[3] Paul, S. and K Sanani, "Reliable Multicast Transport Protocol (RMTP)," IEEE Journal on Selected Areas in Communications, VOL. 15, No 3, April 1997.

[4] Pullen, J., "Reliable Multicast Network Transport for Distributed Virtual Simulation," *Proceedings of the 1999 IEEE Workshop on Distributed Simulation and Real-Time Applications*, 1999

[5] Cohen, D., "NG-DIS-PDU: The Next Generation of DIS-PDU (IEEE-P1278)" 10th Workshop on Standards for Interoperability of Distributed Simulations, March 1994.

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

[7] Defense Modeling and Simulation Office, *High Level Architecture Rules Version 1.3*, U.S. Department of Defense, April 1998

[8] Pullen, J., Myjak, M. and Bouwens, C., RFC2502, "Limitations of Internet Protocol Suite for Distributed Simulation the Large Multicast Environment", The Internet Society, February 1999

[9] Tijms, Henk C.. Stochastic Models: An Algorithmic Approach. England: John Wiley & Sons, LTD., 1994.

[10] Pullen, J.M., *et al*, "A Simulation Model for IP Multicast with RSVP," Internet Engineering Task Force Informational RFC 2490, Internet Society, 1999

[11] Wolfson, H., S. Boswell, D. Van Hook, and S. McGarry, "Reliable Multicast in the STOW RTI Prototype," *Proceedings of the Simulation Interoperability Workshop*, Orlando Florida, March 1997

[12] Bertsekas, D., and R. Gallager, *Data Networks, 2nd Ed.*, Prentice-Hall, 1992, p. 223