Modeling Real-Time Distributed Simulation Message Flow in an Open Network

Dennis M. Moen and J. Mark Pullen
George Mason University
{dmoen, mpullen}@gmu.edu

Abstract

Understanding the characteristics of information flow in large scale real-time distributed virtual simulations (RT-DVS) is important for the development of network services that are able to meet the robust needs of this simulation environment. Being able to quantify these characteristics enables network performance estimations to support the growing demand for use of Internet/Web-based services in large scale RT-DVS applications and interest in simulation interoperability with Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems. This simulation environment requires open network communications protocols that can accommodate the efficient distribution of large amounts of data to many users, which in turn implies the use of many-to-many multicast network services that are not readily available as an open Internet service today. This paper describes results from the study of network information flow in three simulation experiments and presents a supporting analytical model for use in predicting performance of an early implementation of the Extensible Modeling and Simulation Framework Overlay Multicast Protocol (XOM). XOM is designed to support many-to-many multicast for efficient exchange of real-time information among many users in real-time simulations over the Internet. We summarize the architecture and key design considerations of XOM that result from these live simulations studies and our measurements of network traffic and the resulting analytical model.

1. Introduction

Distributed virtual simulations operating across a network in human time generate large amounts of message traffic among the computers hosting the simulations. Without multicasting, this requires many-to-many communications in a dynamic group environment where traffic from each of N computers in the group scales as O(N) message transmissions from each member or O(N^2) total message transmissions in the group [1]. In addition, this simulation environment may not necessarily be homogenous, e.g. each simulator is likely to be different but they dynamically share common simulation objects over time. The result is that simulation objects may have membership in multiple groups with membership changing at different rates and therefore contributing to the high-volume message flow in the supporting network.

RFC2502 [2] describes key networking requirements for distributed simulation that result from the interaction of humans participating in these simulations in real-time. Networks supporting the simulations must distribute large amounts of data within the bounds of the human interaction time, which leads to the need for specific delay bounds. The environment requires many-to-many distributions of data where many senders are sending to many receivers simultaneously. In general networking terms, this environment can be described as a multiparty collaborative environment supporting multimedia applications. The underlying networking environment needs to support a large number of participants dynamically joining and leaving the communicating groups across the myriad of public and private networks that make up the Internet. Because each of these networks is independently managed, the RT-DVS applications cannot solely rely on the Internet to deliver the necessary Quality of Service (QoS) across the Internet, even though QoS mechanisms are beginning to become available on the constituent networks. As a result, networking real-time simulators together has seen limited deployment, and then only in specialized local area networks or on private dedicated networks.

Similar limitations apply to any application that requires group communications across the Internet. In response, members of the research community have proposed an end system approach for supporting multicast across the Internet, where all multicast services are provided by the end system rather than by lower layer network services [3, 4]. Because the overlay provides a place to manage QoS, an RT-DVS application running an end system multicast protocol would also be able to take advantage of the emergence of underlying QoS protocols across the Internet.

The ability to perform many-to-many multicast over an open network is very important to the RT-DVS community and is essential to implementing the Extensible Modeling and Simulation Framework (XMSF) [5]. This framework has been recognized by the Simulation Interoperability Standards Organization (SISO) with the objective of using XML and web based technologies for expanding the user base of RT-DVS. Implementing end system multicast for real-time distributed simulations allows the continued use of open Internet protocols without dependence on the consistency of network policy implementation for multicasting. It also supports the RT-DVS community's effort to move to Web based technologies by providing for multicasting support anywhere on the Internet.

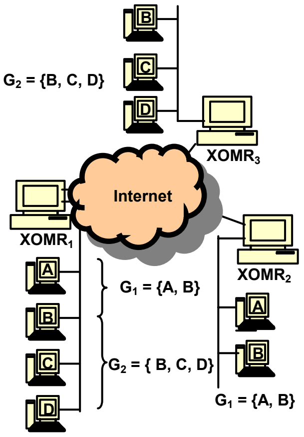
There are many complexities in the RT-DVS environment, including the need for QoS management across a distributed network. No historical data are available to characterize the traffic generated by RT-DVS applications. In order to better understand these complexities, we conducted studies of three simulation experiments in order to first characterize the message flow and then used the results of these studies to aid in the development of an analytical model for use in predicting overlay network performance behavior and the end-to-end system latency.

The analytical model helps establish the regions of feasibility for performance of overlay multicasting in the RT-DVS application environment and provides operational guidance on the level of utilization that should be targeted for the XOM. It also supports the real-time routing algorithms of the XOM in making decisions on overall network optimization during network construction as well as support dynamic overlay network changes during normal operations.

Section 2 of this paper provides a brief background of the current multicast overlay approach and prototype. Section 3 describes our message traffic characterization study approach and Section 4 summarizes the results. Section 5 describes a proposed analytical model that directly results from the traffic studies and is presented in relation to the performance of the current XOM prototype.

2. Background

The need for robust multicast network services to support wide deployment of RT-DVS applications was previously described in our work in building a prototype host-based overlay multicast protocol to support RT-DVS [6, 3]. The XOM provides multicast services over standard UDP or TCP Internet protocols. The XOM is an overlay multicast protocol designed to support many-to-many multicast for real-time distributed visual simulations. The prototype software that implements this is called the XOM Relay (XOMR). From the multicast sender's and receiver's points of view, each XOMR looks like an IP layer multicast router. The XOMR performs as a multicast "relay agent" for any application located on the same subnet as an instance of the XOMR, as shown in Figure 1. Each subnet that participates in the multicast group communication must be supported by a host running a XOMR on the subnet. The XOMR listens to the local LAN for multicast messages generated locally and forwards this traffic to the downstream XOMR(s) according to its multicast tree, by using unicast. Figure 1 presents the concept and indicates the group aggregation efficiency gains by using the overlay multicast. The partner XOMR will then multicast the message to the destination local LAN, and keep on forwarding the message to other XOMRs if necessary. Where high performance is not sought, the XOMR can run on the same computer as the simulation application.



Application B sending implies routing to group $G_3 = \{G_1 \cup G_2\}$

Figure 1. XOMR on a subnet

When an XOMR receives a multicast data message from its local LAN, it will encapsulate it, replicate it, and forward it to the downstream neighbors according to its own multicast tree. When an XOMR receives a message from another XOMR, it will check to see from which source XOMR the message was originally generated. It then checks the locally stored multicast trees for that source XOMR to determine how to replicate the message and forward it to other downstream XOMRs if any. The XOMR also will de-capsulate the message and multicast it to its local LAN.

Our preliminary studies using the performance of the current XOMR prototype give us confidence that the overlay concept is a valid approach for providing multicast services to RT-DVS [7]. We have achieved message throughputs of over 5,000 messages per second while maintaining an overall message loss ratio below one percent. In order to further improve our prototype design and ensure that our testing environment is valid, we undertook a study of real simulation environments. The remainder of this paper summarizes results from these studies. Details of the research work along with an executable version of the current XOMR prototype are available at http://netlab.gmu.edu/XOM/.

3. Characterization study approach

The overall strategy for characterizing the message flow in a simulation environment was to instrument live simulation experiments. The OPNET modeling simulation software [8] tool set was used to aid in data capture and analysis. Multiple data samples were obtained in each of three simulation experiments. These experiments included:

- Wide area network consisting of 2 nodes distributed across the Internet using the XOMR prototype with each node running a Naval Post Graduate School simulation of a maneuvering sea vessel. This experiment allowed for measurement of message flow for a single federate (object).
- Single simulation running on same computer as the XOMR host. This simulation was an Army ground operations maneuver to contact simulation where real-time was actual operations maneuver time or the same as wall clock time. This experiment allowed for instrumentation of an aggregate message flow in a controlled laboratory environment.
- Wide area network consisting of 3 nodes distributed across the Defense Research and Engineering Network (DREN). There were 211 Federates distributed across the network running an urban warfare simulation. This experiment allowed for the study of large volume message flow across an open network.

While all three of these involved real-time simulation environments, there were significant differences in the characteristics among them which proved to be of great value in the overall understanding of the message flow in distributed real-time simulations. Two of the experiments offered unique opportunities as they were across open networks, therefore affording an opportunity to gather data in an environment where the XOM is expected to operate. The one experiment that was conducted in the laboratory allowed for behavior analysis independent of any network layer traffic loading. This measurement experiment therefore offered the opportunity to view pure simulation generated traffic load, unlike the other experiments. This was of considerable benefit in defining an analytical approach.

Capturing performance information of networked simulations depends on complex interactions between the application, the services provided by the operating system of the Host, and the network layer services. Data capture must result in ability to provide detailed, quantitative understanding of these interactions across different network segments. The OPNET module called Application Characterization Environment (ACE) [8] was used to perform this data capture and analysis. The George Mason University C3I Center NETLAB was provided a license to use this product under the OPNET university research program specifically for this project.

4. Summary of characterization results

In this section, we consider the results of the three message traffic studies introduced in the previous section, presenting the experiment design and the specific observations of each.

4.1 Naval Vessel Simulation

The objective in the first experiment was to understand offered network traffic load characteristics in an environment where multicast was employed. The sites involved were two academic XMSF laboratories: Old Dominion University (ODU) Virginia Modeling, Analysis and Simulation Center (VMASC), and GMU C3I Center. The RT-DVS workload consisted of the HLA Federation supported by the Pitch pRTI 1516, which is multicast-capable, and the Web Service Interest Management (WSIM) prototype as described in [9]. The WSIM prototype provided a Web-service based viewer/controller for integrating and distributed simulation object update information from the source simulation applications to the simulation viewer clients at the remote locations. Simulated objects were naval vessels from the NPS SAVAGE library, visualized on a 3D viewer from NPSNET (http://www.npsnet.org). The simulation demonstrated no difficulty sustaining synchronized steaming-in-circles behavior with ship models running at ODU and GMU. The data rates involved were low: less than ten messages per second per site.

Figure 2. indicates the logical tier relationships between the nodes in the network and the two multicast groups used in the message exchange. The tiers are defined as: the Simulation Federate, the WSIM Server (database and web services), Client, Viewer, and the two multicast groups.

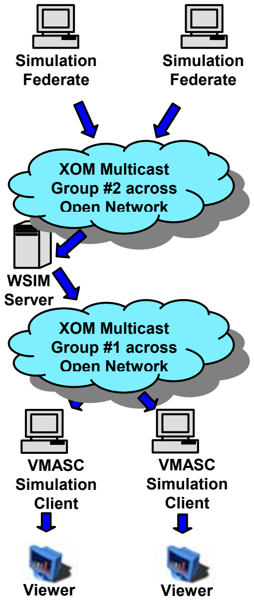


Figure 2. Logical tier relationship for multicast

The logical flow of messages is that the Federates publish to multicast group 2 using the XOM multicast services. The WSIM server listens to multicast group 2 and receives the published updates. The client, using TCP, establishes a tunnel connection to the WSIM Server announcing request for registered information and subscribes to multicast group 1 using the XOM multicast service. The client then listens to multicast group 1 on the local area network subnet in the case of the GMU node and provides the received information to the local viewer. AT VMASC, the client listens to the same multicast group via the XOM services on the VMASC local area network subnet.

Figure 3. provides a view of the measured inter-arrival times of the messages. The horizontal axis represents an individual message arrival in sequence of arrival over time. The vertical axis represents the inter-arrival time of the message in seconds. The figure shows an inter-arrival time centered on .25 seconds which by inverting gives us the 4 message per second rate. The standard deviation for the inter-arrival times is .04228 which supports the observation of a deterministic distribution.

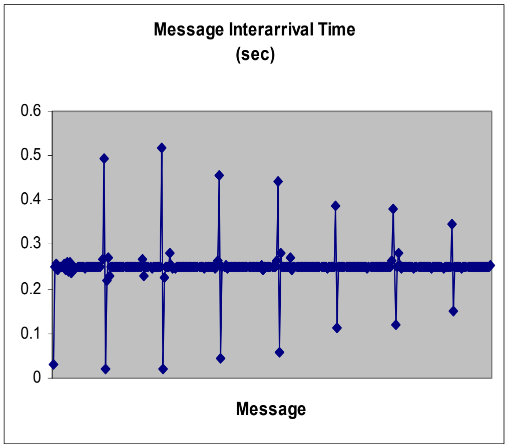


Figure 3. WSIM message inter-arrival time

4.2 Combat operations maneuver simulation

The second simulation experiment was a real-time combat unit operations maneuver simulation consisting of a background static terrain map overlay and approximately 30 mobile/active objects. The general description of the simulation is a friendly force combat unit maneuvering to engage an enemy force deployed in defensive positions. Each object represented a mobile weapons system in the case of the friendly forces and a defensive position with a weapons system in the case of the enemy forces. Real time in this simulation is defined as actual operations wall clock time. This simulation requires many hours of real clock time to completely play out the operations scenario. Therefore, the total scenario was not played to completion but only through initial deployment of friendly forces upon first contact with the enemy forces, which provided a sufficient sample of network traffic.

The test configuration was very simple: the simulation host was connected to the capture agent host using a cross over cable between the two hosts, providing standard Ethernet network capacity of 100 Mb/s. There were no other network connections or other sources of message traffic. There were no observed delays for networking, as expected for this sample network connection approach and relatively low traffic volume.

Figure 4. is a 100 second period of time sample from a large data sample of this simulation and indicates the message throughput as a function of time. This simple is representative of all the data samples taken. The most important observation to be made from this figure is that there are periods of time during which there is a much higher rate of message transfer followed by periods of time with a lower rate of message transfer. We describe this pattern as an

ON/OFF pattern, where ON means a higher rate of message transfer and OFF means a lower rate (not necessarily zero). In this case the ON traffic represented approximately 100 messages per second and the OFF period represented approximately 80 messages per second. Analysis of the detailed data in the capture file reveals an average message inter-arrival time of .011008 sec which translates to an average of 90.9 messages per second or 3.3 messages per second for each object in the simulations. The average message size was 150 bytes.

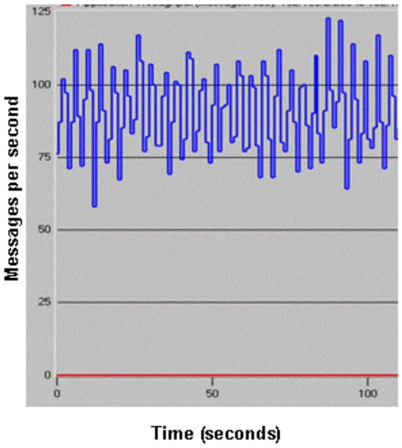


Figure 4. Message throughput

4.3 Joint Forces Command (JFCOM) simulation experiment

As part of a larger research effort funded by the Defense Modeling and Simulation Office (DMSO), George Mason University C3I Center participated in a JFCOM simulation experiment. The participation allowed for the characterization of message flow between elements of a large distributed simulation experiment.

The wide area network had three nodes connected by high speed access (OC3) to the Defense Research and Engineering Network (DREN). The nodes were located in Northern Virginia (Fort Belvoir), Southern Virginia (Suffolk) and Southern California (San Diego). The DREN is an open network environment similar to the open Internet, using standard Internet protocols for routing and message transfer. Distributed across the three DREN nodes were 211 federates connected via HLA Run Time Infrastructure (RTI). The simulation environment contained approximately 110,000 objects. All communications among federates used reliable communications over TCP. No multicast was employed. However, a “pseudo” form of multicast was employed by using intermediate hosts called IMP’s for aggregation of object update messages. (IMP is an acronym used to describe the background clutter used to simulate an urban area.)

In previous experiments operated in the same environment, attempts where made to use standard Internet multicast protocols rather than the intermediate relay host strategy. However these protocols proved too complex and difficult to manage. Reportedly, they did not meet the reliability and other performance needs of the simulation environment. Specifically, the management of group membership did not easily support the nature of the applications need for joining and leaving group subscriptions.

The alternative was implementation of “pseudo” multicast. The update message concept employed was a publish-subscribe model where the aggregate visual space was divided into a fixed grid overlay. A group number was assigned to each grid in the overlay and the objects with in the grid where assigned to the associated group number. A federate subscribes to a group(s) as necessary based on desired view of the overall visual space. Simulation information is then distributed from the head Federate (head IMP) across a 4-layer hierarchical structure of IMP hosts. At each layer there can be several instances of a lower layer host, providing a complex distribution scheme.

The IMPs in the structure are responsible for storing and forwarding element message updates based on group subscriptions of associated Federates. The service provided by this approach makes no assumptions or guarantees about end-to-end latency for the delivery of messages. In the case of this experiment, the users accepted the inherent delay and learned to adjust their interactive response to the delay during the course of running a real-time operational simulation. Multiple traces of data were collected over a two day period of the experiment. The analysis presented here is based on information from one trace of length 8.6 seconds collected at a Layer 4 IMP labeled Blue IMP-163. This trace is very representative of all the traces taken during the course of the experiment.

Inspection of the source message traffic indicated that the dominant traffic source to be the scenario geographic background generator connected to the Head IMP. This message traffic was referred to as “clutter” as it represented the real-time simulation of an urban environment including normal vehicular traffic and movement of pedestrians. The federate was observed to generate a very low volume of message traffic, in most cases consisting only of ACK messages associated with TCP message flow. Figure 5. shows the average message flow rate over time. This figure provides an integrated view of the flow where the flow is from the Layer 3 IMP to the Layer 4 IMP and finally to the Federate. The upper line on the graph is the higher layer flow, including overhead from the message replication scheme. The higher layer rate was nearly twice that between the lowest layer and each of the Federates, as indicated in the tightly coupled set of lower lines on the graph. The general message pattern behavior is representative of an ON/OFF type traffic source and also appears to have uniform ON/OFF periods. This behavior pattern is consistent with the prior simulation experiment observations.

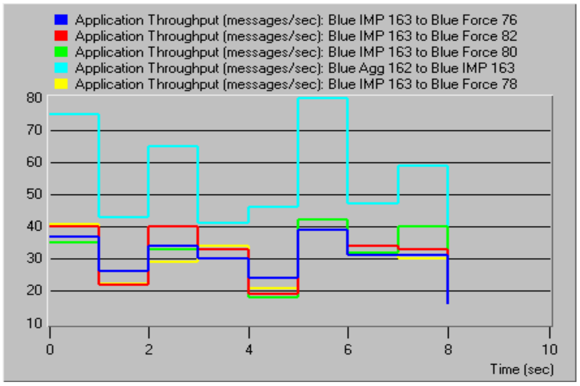


Figure 5. Application message throughput integrated view

5. Analytical model

Two of the three experiments results indicate an ON/OFF traffic pattern as most representative of the traffic flow. These two, the Ops simulation and the JFCOM experiment, represent more realistic scenarios than the WSIM RTI experiment as they were very interactive and involved many Federates that had different characteristics of movement. These experiments represent important observations and provide evidence to support our approach for our proposed analytical model using an ON/OFF model.

Our traffic studies indicate that the simulation environment has relatively deterministic traffic generation rates and that the ON/OFF periods generally have relatively uniform distributions of alternating ON and OFF periods. The other observation of the measured traffic is that the simulations tend to have fixed length messages sizes. This led us to consider an approach that uses an aggregated flow M/D/1 queue model. In this analysis, an assumption is that application data sources have stationary statistics and are independent. Our traffic studies provide evidence that the message traffic behavior is deterministic, validating this assumption. The assumption allows us to use a multiplier, *N*, for summing multiple sources. It also implies that as the number of sources increases, that the traffic load variance will also increase linearly.

Ma [10] and Pitts [11] also have used this approach to simplify the M/D/1 model, based on the idea of flow aggregation. Pitts presents an argument that flow aggregation is possible based on the assumption that the output port, or in our case the access link to the network, has much greater capacity than the offered source rate of message arrivals. In the cases we studied this assumption also was valid. If the capacity threshold is defined as the available service rate *C*, then the rationale used by Ma and Pitts can be used to construct a two-state overflow model as presented in Figure 6. Here the aggregate process is either in the ON state, meaning the message arrival rate exceeds the output service capacity *C* (or defined threshold) and an OFF state where the arrival rate is less than then the service capacity (threshold), but not necessarily zero.

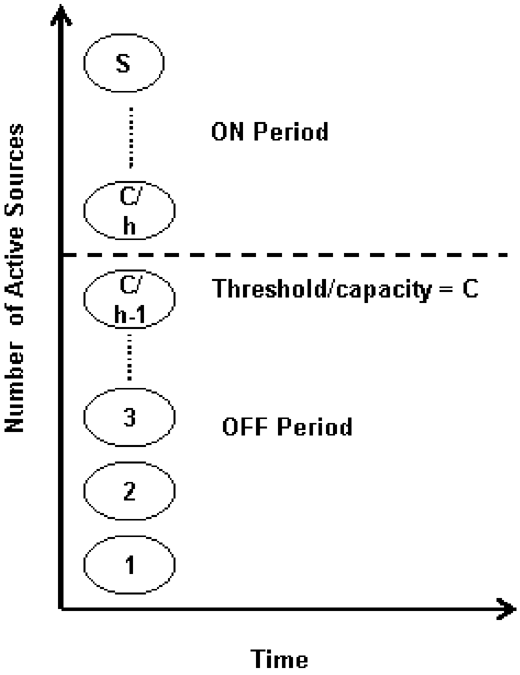


Figure 6. Two state model for threshold capacity

Using the ON/OFF model analytical development details presented in Kang [12], Bianchi [13], Ma [10] and Pitts [11], an approach customized to the distributed simulation application is easily accomplished and results with an expression for the queue performance that gives a measure of performance of an overlay node in terms of the probability of the queue being in overflow. From our traffic studies, we are able to adopt the assumption that the excess rate message arrivals in the ON state are geometrically distributed and that the free periods, or the periods of less than excess rate arrivals, are also geometrically distributed. Then the queue overflow probability is given by Pitts [11] as:

$$P(x)=\frac{\lambda * D}{C-A_{M}} * \left(\frac{1-\dfrac{1}{T_{ON} *\left(R_{ON}-C\right)}}{1-\dfrac{1}{T_{OFF} *\left(C-R_{OFF}\right)}}\right)^{x+1}$$

Here λ =message arrival rate when ON, C =capacity of the access link, A_M =weighted sum of the message rates in the ON and OFF states, T_{ON} =average time in the ON state, T_{OFF} =average time in the OFF state, R_{ON} =the excess message rate in the ON state, R_{OFF} =the excess message rate in the OFF state and D represents the Erlang call waiting formula (probability that a flow will be delayed). The first part of this expression represents the probability that an arriving packet is an excess-rate arrival given by:

$$\frac{\lambda * D}{C-A_{M}}.$$

The second term in the expression represents the geometric progression of the status of the queue and is the probability that the queue exceeds x messages. Since these are independent events, multiplying the terms together results with desired expression for queue overflow probability as presented.

We have validated this analytical approach using our current prototype of the XOM in the laboratory. Under our test conditions, we measured a 5900 message throughput rate with a 1% overall message loss ratio and formulated the probability of overflow model described above. We applied a .7 utilization link factor as a desired threshold and assume this to the rate of average load. Knowing that the Java interface socket has a queuing capacity of 64,000 bytes (approximately 400 messages), then we get the probability of aggregate overflow as presented in Figure 7. The curve represents expected performance at 80% of the XOM measured capacity which results in queue overflow probability of 0.1 at buffer size of 400 messages.

This result also can be used to aid overlay system design in terms of the optimal diameter of the operational overlay network. For example, if we desire to have an end-to-end system message loss rate of 1% and each XOMR in the system represents a loss of 0.1 %, then the system diameter based on message loss is 10 nodes. Further, using the estimated processing delay of a node in the overlay, we also can define the end-to-end path delay of the system. The end-to-end path delay across the overlay is simply the sum of individual node's processing delays in a path of interest plus the inherent network delay due to distance across the underlying network between the nodes. This can be expressed as

$$\mathrm{D}_{\text{path i,j}}=\sum_{n=i}^{j} \mathrm{D}_{\text{node}_{n}}+\sum_{n,m=i,x}^{x-1,j} \mathrm{D}_{\text{Link}_{n,m}}$$

where D represents delay and n and m represent the end nodes of the links along the path from i to j. The first summation is the processing delay at each node along the path and the second summation is the link delay between the nodes along the path**.**

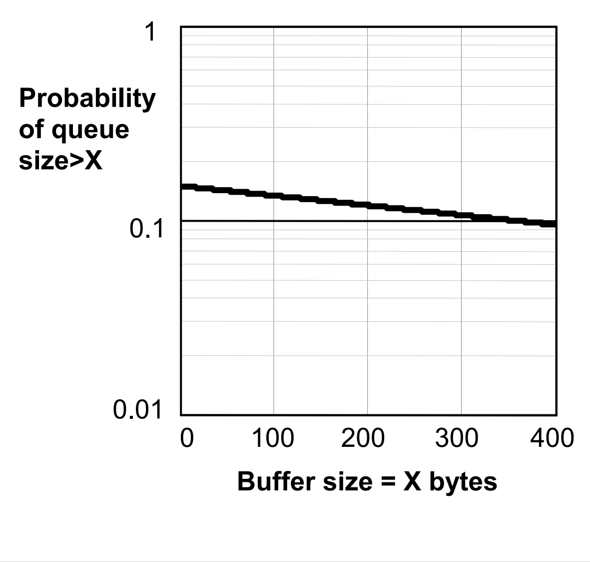


Figure 7. M/D/1 Aggregate probability queue overflow

6. Conclusions and future work

This paper provides results from modeling message load generated by real-time virtual distributed simulations. Three live simulations were studied which resulted in detail characterization at the simulation federate level as well as the aggregated flow of message traffic in a very large simulation with 110,000 federates operating across a private network built over the DREN. The results of these studies enabled the formulation of an analytical model to describe the environment. The approach provides a statistically relevant model that can be directly used for managing and deploying many-to-many overlay multicast network. The analytical model provides the ability to forecast expected performance based on real-time measurement of aggregated message flow through an overlay node.

Many challenges remain in developing overlay multicasting for RT-DVS. Research focus should continue to be on efficiency and scalability as well as overlay management mechanisms that respond to the dynamic nature of underlying open networks as well as the dynamic nature of the application environment. The next steps being considered in continuing our work in overlay multicasting are:

- Validate the approach by using the prototype in a live wide area network
- Establish a testing environment where evaluation of specific routing algorithms can be performed.
- Research and add security features to the prototype for protection from denial of service attacks and implementation of capabilities to support end-to-end information protection features.
- Refine and continue integration with early prototypes of web-services used in RT-DVS.

7. Acknowledgements

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

8. References

[1] Pullen, J., "*Reliable Multicast Network Transport for Distributed Virtual Simulation*", Proceedings of the 1999 IEEE Workshop on Distributed Simulation and Real-Time Applications, 1999, pp. 59-66.

[2] Pullen, J., M. Myjak, and C. Bouwens, "*Limitations of Internet Protocol Suite for Distributed Simulation in the large Multicast Environment*", IETF RFC 2502, 1999.

[3] Moen, D., and J. Mark Pullen, "*Implementation of Host-based Overlay Multicast to Support Web Based Services for RT-DVS*", Proceedings of the Eighth IEEE International Workshop on Distributed Simulation and Real Time Applications, 2004, pp. 4-11.

[4] 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, August 2001, pp. 55-67.

[5] 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.

[6] Moen, Dennis, and J.M. Pullen, "*Enabling Real-Time Distributed Virtual Simulation over the Internet Using Host-based Overlay Multicast*", Proceedings of the Seventh IEEE Workshop on Distributed Simulation and Real-Time Applications, 2003, pp. 30-36.

[7] Moen, D., and J. Mark Pullen, "*Performance Evaluation of the XMSF Overlay Multicast Prototype*", Proceedings of the Simulation Interoperability Standards Organization (SISO) Spring Interoperability Workshop, paper 05S-SIW-024, San Diego, CA 2005.

[8] OPNET Technologies, Inc. Web, http://www.opnet.com/products/modules/ace/home.html 1/20/2005.

[9] Morse, K., R. Brunton, J.M. Pullen, P. McAndres, A. Tolk, and James Muguira, "*An Architecture for Web Services Based Interest Management in Real Time Distributed Simulation*", Proceedings of the Eight IEEE Distributed Simulation Workshop on Real Time Distributed Applications, 2004, pp. 108-115.

[10] Ma, John, "*Physically-Based, Two-State Markov Process Model for Converged Application Traffic with Extension to Burstiness and Self-Similarity*", OPNET2004 Conference Posture Paper, pp. 1-10.

[11] Pitts, J. M., and J. A, Schormans, Introduction to IP and ATM Design and Performance, Second Edition, John Wiley & Sons, Ltd, West Sussex, England, 2000.

[12] Kang, S.H.; Sung, D.K., "*Two-state MMPP Modeling of ATM Superposed Traffic Streams Based on the Characterization of Correlated Interarrival Times*", Global Telecommunications Conference, 1995. IEEE GLOBECOM '95, Volume: 2, 13-17 Nov. 1995, pp. 1422 - 1426.

[13] Bianchi, Giuseppe, Antonio Capone, and Chiara Petrioli, "*Throughput Analysis of End-to-End Measurement-Based Admission Control in IP*", Proceedings of. IEEE INFOCOM '00, 2000, pp. 1461-1470.