A Distributed Interactive Simulation Intranet Using RAMP, a Reliable Adaptive Multicast Protocol

W. Garth Smith, Alex Koifman

ABSTRACT
A dynamic, heterogeneous, multicast environment has been fielded as a simulation Intranet over a wide area network (WAN) for distributed interactive simulation (DIS). By replacing the usual UDP-based unreliable broadcast and TCP-based reliable unicast mechanisms with a single multicast transport protocol (RAMP), the diverse goals of enhanced scalability, reliability (as needed), and efficient operation outside of LAN environments are achieved while still meeting the DIS stated maximum latency requirements. To further support scalability, hierarchies of reliability requirements have been established for DIS protocol data unit (PDU) types. The hierarchies are based on the reliability requirements of individual entities, and are used to dynamically select between RAMP's reliable and unreliable transport modes. By restricting use of reliability to only times when it is really needed, control channel traffic required for reliability is greatly reduced, furthering both scalability and performance. The viability of this approach has been demonstrated through simulation exercises conducted via RAMP-enabled PDU transmissions across an Intranet comprised of multicast-capable routers and commercial WAN segments interconnecting five geographically disperse TASC simulation labs. RAMP also provides automatic fragmentation and reassembly of large packets, which facilitates transmission of DIS packets (such as environmental PDUs) that exceed the size limitations imposed by the Ethernet specification. The benefits to the DIS paradigm for a heterogeneous reliability are also shown to support the proposed next generation simulation initiatives under way with the object request broker (ORB) based high level architecture (HLA) efforts. Initiatives by the Object Management Group (OMG) to support multipoint-to-multipoint communications with the next generation of ORBs is currently underway. TASC has developed methods for embedding RAMP under IONA's ORB implementation (ORBIX) that support multipoint-to-multipoint communications, and is in the process of submitting recommendations for incorporating multicast protocols to the OMG.

Keywords: Communications Architecture, Communication Network Long-Haul Network (LHN), RAMP, Multicast Networks, Reliable Services, Wide Area Network (WAN)

1.0 INTRODUCTION

An internally-sponsored TASC research effort recently developed and demonstrated a network infrastructure for Distributed Interactive Simulation (DIS) between multiple, geographically disperse corporate sites. The primary goal of this Intranet [1] is providing fully synchronized depictions of virtual world representations amongst the several simulation labs. The distributed simulation Intranet interconnects TASC sites in Fort Walton Beach, Florida, Orlando, Florida, Reading, Massachusetts, Reston, Virginia, and San Antonio, Texas (see Figure 1).

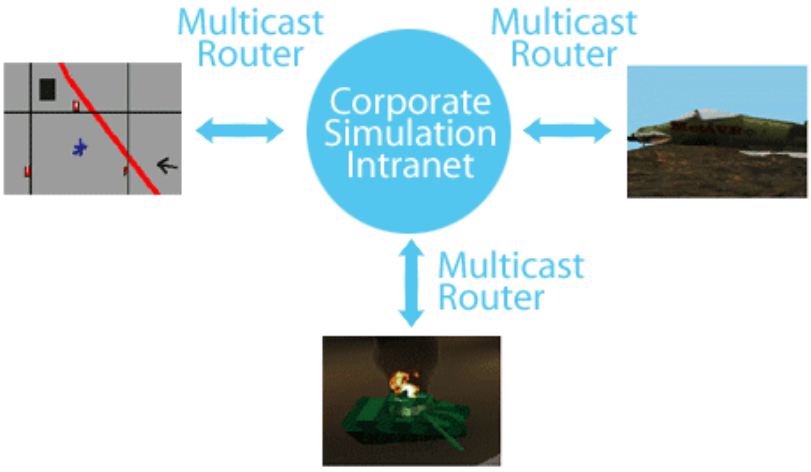


Figure 1. A corporate simulation Intranet for multiplayer virtual environments.

- Linking these sites required overcoming a number of obstacles, including:

- The adverse impact of simulation network traffic on corporate LANs DIS's stated maximum network latency among entities
- Heterogeneous WAN segments with communication bandwidths ranging from 56 kilobits/second to 1.544 megabits/second
- Packet loss and out-of-order arrival that can result from network congestion and alternate routing typical of Internet environments
- Retrofitting the multicast approach with the existing base of DIS models, where a wide range of both commercial and noncommercial network connectivity libraries have been used

We also wanted to meet the additional objectives of:

- Supporting rapid inclusion of other sites in the DIS network
- Providing scalability to support large numbers of entities
- A simulation Intranet that would scale to the Internet
- Ensuring low cost

2.0 LATENCY SUFFICIENCY

Real-time requirements within a DIS environment mandate that an entity must be able to communicate its state information to all other entities requiring the information within a timeframe such that human operators perceive a reasonable approximation of reality in the immersive environment interactions.

Latency sufficiency specifies the bounds on that time frame in terms of a maximum acceptable delay between host processors connected via some network topology.

The TASC WAN infrastructure comprises variable bandwidth data line connections spanning geographically disperse sites and ranging from full T1 (1.544 megabits/second) to 56 kilobits/second. Acceptable transmission times for point-to-point communications from existing standards were compared to actual observed latency between network hosts. Initial benchmarks were determined from the Communication Architecture Requirements (CAS) document Standard for Distributed Interactive Simulation draft 1278.2 IEEE [2], which provides details of acceptable latencies for given types of simulations. This standard was recently balloted and is currently undergoing a number of modifications. The CAS document indicates that crewed simulators have minimal latency tolerances between 100 to 300 milliseconds and computer-generated forces have a tolerance of 500 milliseconds. Latency sufficiency is the upper bound of acceptable time of travel for a PDU between a DIS transmitter and receiver entity.

An experiment to determine the latency across the network was conducted using the UNIX ping facility as a simulated DIS simulator. Ping can be configured to act like a DIS simulator in that the packet size, number of packets, and frequency of packet transmission between hosts spanning the WAN can be set to replicate the network traffic of a simulation. The UNIX command is depicted in Figure 2.

Note that the packet size argument is set to 178 as ping adds its own 8-byte time stamp header per packet. This enables the ping packet size to be 186 bytes, which replicates the DIS Entity State version 2.0.4 PDU byte length. We assume symmetric round-trip transmission time when evaluating the results of ping. Hence, one half of the total average round trip time resulting from ping is used to determine a heuristic for packet latency. In general, one should also compute histograms for the maximum and minimum latency to account for dynamic bandwidth that is typically compounded by sharing network resources between corporate and DIS traffic. The ping simulator is then run at time intervals that match the packet rates per second for the given transmission rates found for a given number of entities. Latency results then mirror the expected PDU transmission rates over a given time interval. We assume a uniform distribution over the time interval for the packet transmissions. These steady-state conditions provide a general heuristic for determining network behavior. Although these metrics do not fully bound the network behavior, they prove satisfactory in practice.

The resulting average latency among sites then is 80 milliseconds and the minimum latency is 30 milliseconds. These values fall within the most stringent requirements recommended by the CAS [2] document for one-way communication. The resulting values are also below the threshold for computer-generated forces for acknowledgment-based transaction protocols for the case of reliable transmissions.

3.0 SIMULATION NETWORK PROTOCOL CONSIDERATIONS

A large body of the existing DIS simulations in industry use the SimNet [3] paradigm of a UDP/IP broadcast transport layer for communicating entity-based network traffic. It is well known that this model forces all host CPUs to inspect all the network traffic, causing interrupts at the operating system level for all packets. These issues are exacerbated in the case of WAN based communications. Imagine all hosts attached to a WAN receiving broadcast transmissions from a single DIS simulation! For this reason, many WAN routers are programmed to not forward broadcast traffic.

Multicast network transmissions provide the ability to specify groups or ranges of receivers, meaning that not all the host CPUs need receive the network traffic. As an added benefit, many network cards provide the ability to filter upon IP address information in the multicast packet so that nonrelevant packets can be discarded without requiring an interrupt to the operating system. These attributes were the basis for choosing multicast as the communication layer for simulation transmissions.

Major limitations of the IP multicast, however is the lack of support for reliable, ordered, and yet simultaneously scalable transmissions. These features become critical in the more hostile environment of

a WAN. Specifically, a frame relay environment is more conducive to packet collisions than an Ethernet LAN. Also, fully reliable transmissions do not scale well for large simulation exercises.

A more significant limitation at the beginning of the research effort was the lack of commercial vendors providing IP multicast-capable routers for WANs. Two crucial pieces of technology were identified as potential solutions to reliability issues and the lack of multicast-enabled routers. A heterogeneous reliability protocol for the transport layer was identified in the TASC- and ARPA developed Reliable Adaptive Multicast Protocol (RAMP) [4]. Until commercial vendors began shipping the new multicast-based routers, the software-based mrouted [5] routing technology was identified as a WAN multicast router. The mrouted application is part of the multicast backbone (MBONE) [6] technology which is an array of freeware-based applications meant to bring true multicast applications to the Internet. Others have used the MBONE for unreliable IP/multicast based DIS experiments over the Internet [7].

4.0 RAMP OVERVIEW

RAMP was initially described in IETF RFC 1458 [8] and is used within TASC for collaborative interactive and image transfer applications as well as distributed simulations. An effort is underway to introduce RAMP into the commercial DIS software community. RAMP provides a single simulation process the ability to span a range of sender and receiver reliable and unreliable delivery modes, based on the type of data being transmitted. Addressing these issues at the network level rather than the application level are crucial for efficient use of simulation resources.

In reliable mode, depending on the nature of PDU type, retransmissions may not be appropriate. If the round trip time exceeds the pre-established latency sufficiency criterion, retransmissions for Entity State PDUs may not be appropriate. Or, the transmission frequency of the data type (i.e., entity state PDU) may obviate the need to transmit reliably. However transmissions such as detonations, radar illuminations, and resupply DIS PDUs occur less frequently and do provide for graceful recovery for transmission loss. Given the nature of their importance as a significant and infrequent event, they must be performed reliably. Making all transmissions reliable would overload the network and hence we take the approach of having a heterogeneous dynamic reconfiguration of socket connection based on a criterion of necessity of reliability.

RAMP is a transport layer multicast protocol that operates over network layer multicast protocols such as IP/multicast. It provides a standard method for reliable point-to-multipoint transmission. RAMP guarantees reliable and orderly delivery to all multicast recipients using a negative acknowledgment approach to minimize return (control) traffic. RAMP can be described as a connection-oriented, reliable stream service. However, as message boundaries are also preserved, both datagram and stream-style interfaces are supported.

Unlike a number of multicast transport protocols that are only receiver reliable [9], RAMP is *fully* reliable. That is both sender *and* receiver reliable. With full reliability, applications are notified whenever a sender or a receiver fails. This is highly desirable for collaborative interactive applications where each users application acts both as a sender and a receiver to a single multicast group. Here, state information about the presence or absence of individual users must be maintained to support operations such as resource recovery (e.g., tokens) resulting from individual network or application failure.

RAMP features include timely notification of receiver failure to the sender (at most a few seconds), as well as timely notification of sender failure to all receivers. RAMP provides fast joining and leaving of groups, where participating receivers can leave a group or new receivers can join a group at any point in the session. Mid-session dynamic modification of group membership allows individual instances of distributed applications, such as shared whiteboards, to be initiated and terminated at different times without requiring lengthy resynchronization. As RAMP maintains explicit group membership at the sender, transmission can be terminated immediately when the last receiver leaves the group rather than continuing to consume valuable network resources.

RAMP also supports mixed reliable and unreliable transport in that some of the receivers can elect not to set up control channels or send retransmission messages. This feature is useful for multicast transmission of hierarchically encoded data sets, where partial loss by some of the receivers still yields useful information to those receivers. For example, an image sent from a ground station to an archive must be performed reliably; however, other listeners can elect to receive reliably only the lower-resolution levels. If partial information from the other levels becomes available to those receivers, it can be used to increase the quality of the imagery at the corresponding positions in the image. The protocol also provides sender-based reliability, where the sender can elect to send data either reliably or unreliably. Sender-based unreliable transport under RAMP is similar to UDP transport, with RAMP providing the added functions of segmentation and reassembly of large messages. As shown in Figure 3, RAMP is a transport protocol layered on IP/multicast.

Although the functional model for RAMP is somewhat similar to TCP, TCP establishes a full-duplex reliable connection between two endpoints, whereas RAMP establishes a simplex (one-way) reliable flow between the *sender* and the *receiver group* (Figure 4).

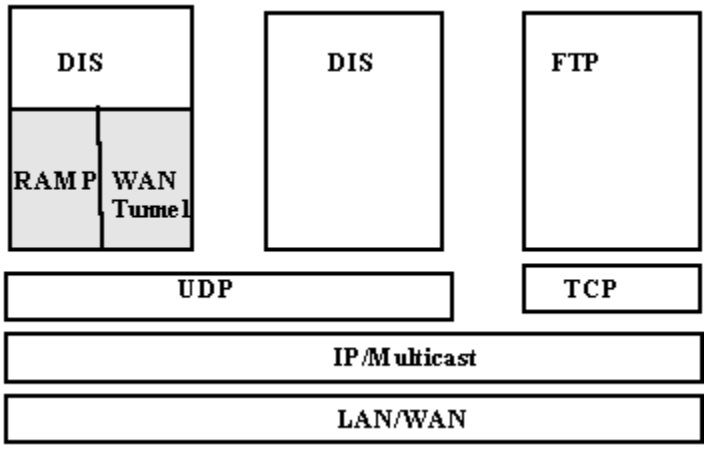


Figure 3. Comparison of the RAMP protocol stack with traditional broadcast DIS and FTP.

Although RAMP *does* provide limited (single segment) data delivery from a receiver back to a sender in the form of piggyback messages, the reverse path is primarily intended for control information.

RAMP supports both a simplex and duplex model for multicast communications. Although bus based networks such as Ethernet do in fact support full-duplex interconnectivity, the RAMP architecture in no way precludes an application from establishing a full-duplex flow. An application that requires a full-duplex data flow need only create two RAMP flows: a forward flow and a reverse flow. Each flow functions independently and can provide a different quality of service (QoS).

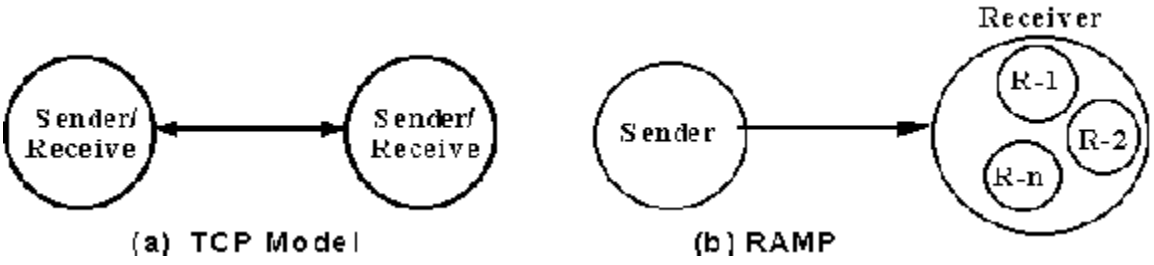


Figure 4. Two protocol models.

RAMP flow consists of segments, in which each consecutive segment has an increasing sequence number. Reliability is achieved using a negative acknowledgment scheme; a receiver notifies a sender immediately upon detecting a gap within the received sequence. Explicit acknowledgments are only required during the initiation and termination of connections.

4.1 Unreliable Delivery Under RAMP

RAMP provides two types of unreliable data delivery to support applications in which unreliable delivery is acceptable and appropriate. The first type is the more familiar unreliable connectionless delivery model in which the sender and all receivers operate unreliably. It is appropriate for voice and video applications. Here unreliable data delivery is similar to UDP, with the addition that RAMP supports larger messages (those greater than 64 kilobytes) through segmentation and reassembly. The receivers do not send acknowledgments or retransmission requests to the sender, and the sender does not accept or process acknowledgments or retransmission requests from the receivers.

The second type of unreliable delivery involves a somewhat unique model in which the sender supports reliable delivery, yet some (or all) of the receivers operate in an unreliable mode. This is appropriate, for example, for image delivery services in which certain receivers (such as image archive servers) require reliable delivery of all image data, yet other receivers can accept some data loss, such as the loss of higher resolution data in hierarchically encoded images. The second type of unreliable delivery allows a single multicast sender to simultaneously support both reliable and unreliable receivers with a single data feed.

In RAMP, both the sender and receiver can freely switch between reliability modes. The sender moves between reliability modes by toggling the reliability flag (bit) in its messages, indicating whether receivers are allowed to issue control messages to the sender. Receivers switch between modes by simply processing or ignoring lost messages.

Data messages are sent from sender to receivers. Messages over 8000 bytes are fragmented into segments by sender and reassembled by receivers. Each segment contains a variable size header and a maximum of 8000 bytes of data. Figure 5 illustrates the data flow segment format.

The first 8 bytes of the RAMP message headers are intentionally identical to the UDP message header so that completely RAMP-compatible messages can be constructed using UDP yet avoid kernel programming during development. Data message types include Connect, Accept, ACK, Idle, Close and Data.

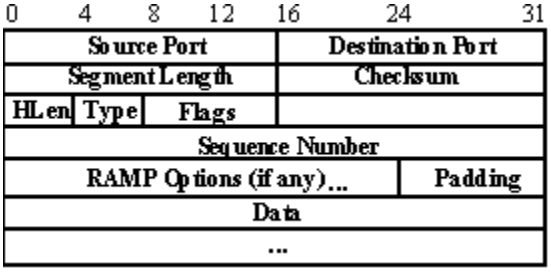


Figure 5. Data flow segment format in reliable delivery.

4.2 RAMP and DIS Network Library Integration

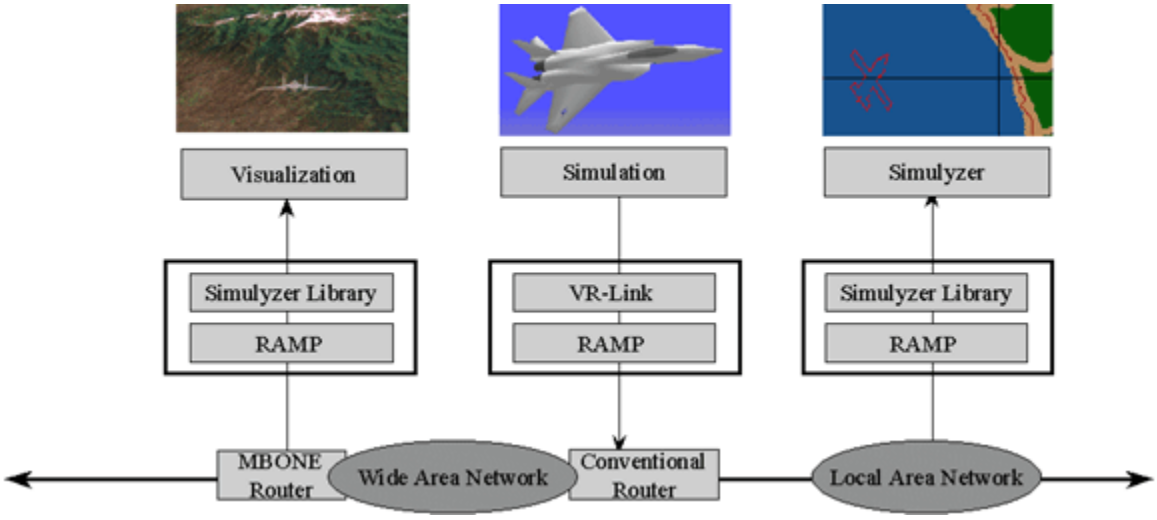


Figure 6. RAMP-enabled commercial and noncommercial libraries providing WAN- and LAN-based heterogeneous reliability for DIS applications.

Currently, TASC uses both commercial and internally developed software libraries to provide DIS network connectivity for simulation and visualization applications. The goal was to augment these existing libraries such that RAMP was treated as another run-time option for any given application. Both the MaK Technologies VR-Link toolkit [10] and the TASC Simulyzer [11] DIS network connectivity libraries were augmented with RAMP.

RAMP's Berkeley sockets style C application programmers interface (API) (see section 4.3) was integrated into both the VR-Link and Simulyzer C++ libraries to support simulation over LANs and WANs. Initially, we created a library for a half-duplex connection-oriented RAMP socket. Using this library, we added a library for a full-duplex connectionless RAMP socket. This full duplex library was then integrated into both network connectivity libraries. In the case of Simulyzer, direct integration was readily possible as direct access to the source code was available. In the case of VR-Link, integration was also readily performed even though access to only the header files and object code was possible. The C++ virtual methods in the VR-Link NetSocket class enable users to define the transport layer by writing their own send and receive functions. The resulting Intranet software infrastructure is depicted Figure 6.

Simulyzer provides an engineering-level visualization capability. This RAMP-enabled DIS Stealth application has been executed among the TASC simulation labs. Anecdotally, the minimum latency requirements are confirmed in that no visual anomalies occurred during the real-time simulation experiments conducted to date. A major limitation of the mrouted software routers is their mimicry of multicast by establishing tunnels across the WAN. These tunnels function as point-to-point transmitters and, thus, are not true multicast routers. The resulting additional network traffic leads to relatively small but valuable simulations exercises. Upon upgrade of the corporate network (TASCnet) to true multicast-capable hardware routers and uniform T1 capacity, exercises across the WAN scale to larger numbers of entities. The performance of LAN-based simulations with unreliable RAMP is similar to traditional multicast implementations in terms of the maximum number of total entities that can be simulated.

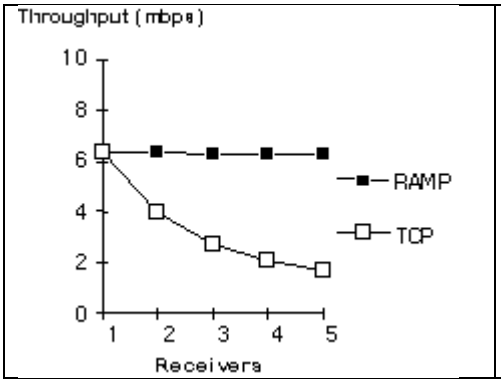
4.3 Application Program Interface

The API for RAMP is similar to the BSD/UNIX API for TCP. The following function calls are provided to manage RAMP sockets:

- *rsocket - creates a RAMP socket.*
- *rbind - binds a name to a socket.*
- *rlisten - listens for connections on a socket.*
- *rconnect - initiates a connection on a socket .*
- *raccept - accepts a connection on a socket.*
- *rsend/rsendto - sends data from a sender t o a receiver group or sends a piggyback data from a receiver to sender(s).*
- *rrecv/rrecvfrom - receiv es a RAMP message. Also, allows a receiver to accept a connection.*
- *Recv allows receipt of data in the stream or datagram format.*
- *rclose - closes a RAMP connection. If issued by a sender, closes a RAMP connection to a receiver group (to all receivers). If issued by a receiver, closes a connection between the receiver and the sender(s).*
- *rgetsockopt - gets the value of the various socket options.*
- *rsetsocktopt - sets various socket options. Supports most T CP/IP options. Additional RAMP options are Idle or Burst mode selection, reliable or unreliable delivery, checksum on/off and user defined options.*

4.4 Performance Over Ethernet

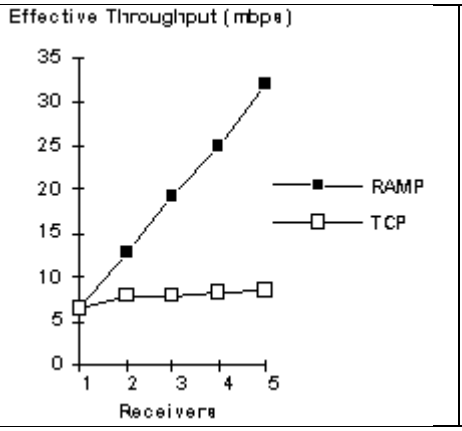
Measurements were made of TCP performance over Ethernet to provide an absolute evaluation of RAMP's performance. As shown in Figure 7, when TCP is used to provide a reliable multicast like service using repeated transmissions, the throughput to each receiver is reduced by 1 over *n*, where *n* is the number of receivers.



This expected result occurs as the total bandwidth for the network is fixed and must be shared by each of the n receivers. In contrast, the throughput to each receiver remains relatively constant when reliable multicast is used, also as expected. Figure 7 primarily illustrates that RAMP is at least as good as TCP as a unicast protocol, and that RAMP behaves properly as a reliable multicast protocol.

Figure 7. Comparison of RAMP and TCP throughput over Ethernet.

An alternate presentation of the results is given in Figure 8. RAMP vs. TCP effective throughput over Ethernet, in which the effective throughput (sum of throughputs to each receiver) is given as a function of the number of receivers for both TCP and RAMP.



Clearly there is a substantial advantage to using RAMP over TCP, even for a receiver group as small as two, with the advantage increasing quickly as the number of receivers grows. For example, to achieve the same effective throughput as RAMP with 5 receivers (32 megabit/second), TCP would need to be running over a 50 megabit/second LAN rather than the 10 megabit/second Ethernet used here.

Figure 8. Comparison of RAMP and TCP Effective Throughput over Ethernet

4.5 Reliability Hierarchies

A critical requirement for the success of the implementation of this protocol and its myriad capabilities for transmission is the need to abstract reliability notions from the user. Under development are parameter-driven reliability hierarchy files. These parametric options establish criteria for the types of transmissions that are to occur during a simulation exercise. It is interesting to note that in the evolution of the DIS protocols, heterogeneous reliability was an initial requirement: The DIS 2.0.3 standards [12] refers to the Issuance of the Collision PDU: The collision PDU shall be issued by using a real-time, reliable, multicast communication service. However, in the 2.0.4 standards [13], reference is made to the same PDU by stating, the collision PDU shall be issued by using a best effort multicast communication service. Notice that in both protocol versions mentioned, reliability was not a requirement for the Entity State PDU. Hence, the standard was attempting to account for heterogeneous reliability. It is the hope of the authors that the DIS standards committee would re-embrace the notion of heterogeneous reliability now that a potential solution in RAMP exists for mixed reliability in distributed simulations. Table 1 is an example of the parametric reliability hierarchies currently being investigated. These parameter files would be user-configurable and set at run time and in general the user would need not be aware of its details.

DIS PDU Type	Reliability Requirements	Duration
detonation	fully reliable	entire exercise
collision	fully reliable	entire exercise
environmental	fully reliable	entire exercise
entity state	conditionally reliable	while < 1000 meters apart

Table 1. DIS packet types and their proposed associated dynamic reliability requirements.

5.0 RAMP ENABLED ORBS TO SUPPORT HLA

Current next-generation simulation technology involves supplanting the DIS paradigm with the Common Object Request Broker Architecture (CORBA). The high level architecture (HLA) initiative is planning to use an object passing model that uses TCP/IP as the transport protocol. As the limitations imposed by full connection based protocols to achieving large scalable exercises are well known, initiatives are under way to investigate multicast enabled object request brokers (ORBs).

Incorporating RAMP within a CORBA framework is a significant challenge. Although the Object Management Group [14] (OMG) specifies that CORBA should support the use of alternative transport protocols, in practice ORB providers have generally restricted their implementations to a particular transport protocol; specifically, TCP. In order to support multicast, TASC has developed and implemented a novel approach to circumvent CORBA's lack of support for multicast protocols. The approach layers RAMP underneath the same interface definition language (IDL) API that is used for unicast communications, both ensuring CORBA compliance and allowing for transparent access to both unicast and multicast services. Client applications can be written to use a single API, regardless of the communication protocol (unicast or multicast) being used. Clients without access to RAMP receive data using the ORB's unicast protocol, whereas clients that have access to RAMP receive the same data via a multicast protocol. Although the current service supports only point-to-multipoint delivery of image data, we have begun the process of generalizing and extending this approach under IONA Technologies' ORBIX CORBA environment so that all communications services, including multipoint-to-multipoint (peer-to-peer) will be afforded this capability.

6.0 SUMMARY

RAMP is currently in the request for comments (RFC) phase as part of an effort to achieve general acceptance of the protocol. It has proven itself in a number of internal and external implementations for multi-media, image transfer, and simulation. The novel approach of a multicast heterogeneous reliability coupled with hierarchies of reliability for types of simulation transactions has promise for making large scale high fidelity virtual worlds possible. The performance assessment reveals that RAMP is an effective protocol for reliable transmission even when used as a unicast protocol, that RAMP scales linearly for reliable multicast operations, and that RAMP provides quite reasonable performance as a concast protocol making it suitable for collaborative applications. RAMP has been implemented for IP/UNIX workstations and has been tested over both Ethernet and ATM networks.

BIBLIOGRAPHY

[1] "Enter the Intranet," pp. 64, 65, *The Economist*, January 13, 1996.

[2] The Communication Architecture Requirements (CAS) document "Standard for Distributed Interactive Simulation draft 1278.2 IEEE."

[3] A. Pope, BBN Report No. 7102, The SIMNET Network and Protocols, technical report, BBN Systems and Technologies, Cambridge, MA, July 1989.

[4] A. Koifman and S. Zabele, "RAMP: A Reliable Adaptive Multicast Protocol," Fifteenth Annual Joint Conference of the IEEE Computer and Communication Societies, San Francisco, CA, March 26-28, 1996.

[5] http://www.best.com/~prince/techinfo/misc.html#ftpsites

[6] http://www.research.att.com/mbone-faq.html

[7] M.R. Macedonia, M.J. Zyda, D.R. Pratt, P.T. Barham, S. Zeswitz, "NPSNET: A Network Software Architecture for Large-scale Virtual Environments," *Presence*, Winter 1994.

[8] R. Braudes, S. Zabele, "Requirements for Multicast Protocols," *RFC 1458*, May 1993.

[9] H.W. Holbrook, S.K. Singhal, D.R. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation," to be presented at SIGCOMM 95.

[10] J. Morrison, "The VR-Link Networked Virtual Environment Software Infrastructure," *Presence*, Spring, 1995.

[11] http://www.tasc.com/simweb/papers/simulyzer/simulyze.html

[12] Institute for Simulation and Training, Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, Version 2.0, Third Draft, University of Central Florida, Orlando, FL, May 28, 1993.

[13] IEEE Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications IEEE Std. 1278-1993 May 12, 1993.

[14] http://www.omg.org