As published in *Proceedings of the 1999 IEEE Distributed Interactive Simulation and Real-Time Systems Workshop*

Reliable Multicast Network Transport for Distributed Virtual Simulation

J. Mark Pullen
Dept of Computer Science and C3I Center
George Mason University
Fairfax, VA 22032
mpullen@gmu.edu

Abstract

This paper addresses the need for reliable multicast (RM) network support for the class of distributed virtual simulations (DVS) that operate with human participants who experience a virtual representation in real time, for example DIS and real-time HLA federations. The need for RM in DVS is explained. The state of reliable multicast networking is explained and typical mechanisms for providing it are described. There is no general solution to RM; all existing RM protocols take advantage of some aspect of their application domain for successful function. Aspects of the DVS domain that lend themselves to providing RM are described, along with an approach to RM for DVS that has been prototyped successfully in the Selectively Reliable Transport Protocol (SRTP). The workings of SRTP are described, including its application to DIS and real-time HLA. SRTP enables efficient use of network-level multicast, and as such represents an advance over the multicast emulation that is used in the Run Time Infrastructure (RTI) created for the Synthetic Theater of War (STOW). In conclusion, ongoing work and potential future improvements to SRTP RM are described.

1: Introduction

This paper addresses multicast network support for the class of distributed virtual simulations (DVS), meaning those simulations, distributed over a network, that operate with human participants who experience a virtual representation in real time [20]. Distributed Interactive Simulation (DIS) [7] in an example. A class of federations providing real-time, human-in-the-loop distributed operation under the High Level Architecture (HLA) for simulation [5], such as the Synthetic Theater of War (STOW) [1], also can be categorized under DVS. Network support for large configurations of these simulations is a challenging problem because of the real-time requirement and the scope of its distribution [13, 21].

Network-based distributed simulation characteristically generates large amounts of message traffic among the computers supporting elements of the simulation. In the general case this requires many-to-many communication among a group of computers conducting some aspect of the simulation. For N computers in the group this scales as O(N) transmissions from each member, or $O(N^2)$ overall, severely constraining options for scaling up the number of participating computers.

Network-level multicast provides an attractive way to bring the distributed simulation problem down to O(N). In a multicast network, the switching elements of the network replicate those packets that have group (as opposed to unicast or single machine) destination addresses, so that every computer participating in the group receives a copy of each packet sent to the group by any other member computer. In local area networks (LANs) there is usually a broadcast paradigm (all-to-all) that can substitute for multicast at scale of tens to hundreds of computers. However in Wide area networks (WANs) multicast networking becomes a virtual necessity for real-time distributed simulation as N increases beyond roughly ten, because of the prohibitive expense of providing network support at scale of $O(N^2)$.

DIS typically has been implemented over broadcast networks, although the potential benefits of multicasting are recognized [8]. The HLA which is superseding DIS contains refinements that reduce redundancy seen in DIS data transmissions. However when used for DVS HLA still shares the basic requirement for many-to-many multicast because it implements the same basic paradigm as DIS. Under the HLA, distribution is achieved using a "Run Time Infrastructure" (RTI) which supports communication among participating computers [2,4]. Multicast network support in the largest-scale RTI implementation reported to date is a unicast-based "emulation of a true multicast protocol" that "is not recommended for heavy message users" [22]. Developers of that network support recognized that, even at only eight network nodes, it carried an expensively increased level of network traffic. However they had no alternative because a workable solution implementing reliable multicast was not available. Because of this lack of scalability, the solution of [22] would have been completely untenable in the large-scale simulation environment envisioned by [21].

2: Reliable Multicast

Distributed simulation depends on the interconnecting network to deliver messages without loss. The interactive form of distributed simulation further requires low latency (delay) in delivery in order to maintain an accurate representation of reality in real time for the participating humans. No network is perfect, thus it is inevitable that some messages will be lost, in which case results obtained in the various simulators will not be entirely consistent. The inconsistency must be held within acceptable bounds. Doing this requires a means to distribute state data for every simulated object with acceptable reliability. To achieve this distribution with good efficiency will require some form of reliable multicast in the supporting network.

2.1: Reliable Multicast Mechanisms

The need for reliable multicast (RM) derives from the value of using the commercially dominant Internet Protocol (IP), which has the property that network packets can be lost because of noise-induced data errors and dropped by network routers when their packet queues overflow. The fact that IP delivers packets only on a Best Effort (BE) basis might seem to be a shortcoming, but the marketplace has found otherwise. Customarily reliability is provided by pairing IP with the reliable transport Transmission Control Protocol (TCP), a layering that has proved to be highly effective in the Internet. Resulting TCP/IP networks are robust and practical. This has led to their commercial dominance, making IP the network protocol of choice for distributed simulation for reasons of economy and sustainability. The benefits of IP are even more clear when the network supporting the simulation is to be shared by other functions, as virtually every network application is supported by TCP/IP today. However, TCP has a point-to-point model of communication where each endpoint is defined as the computer's IP address concatenated with a locally unique "port number" where the unique address+port identifier is used to multiplex multiple TCP streams to the same computer. This means the TCP connection is always point-to-point, thus it cannot support multicast. Furthermore TCP has proved impractical to modify for multicast because it works by sending an acknowledgement (ACK) containing an identifying sequence number (SN) for every packet correctly received. In a multicast implementation, this would result in "ACK implosion" that would overwhelm a sender with ACKs from the many receivers in its group.

Reliable multicast is possible. However results to date indicate there will not be a "one size fits all" reliable transport protocol for multicast, comparable to TCP. In the multicast case, no single mechanism is available to take on the role of the ACK in TCP. Instead there are multiple mechanisms that can be combined to achieve RM under specific circumstances, but no general solution. [12] and [14] give an overview of the state of RM protocol development. Some important RM mechanisms are:

Continuous update of state data at frequent intervals: Real-time multimedia streams (audio and video) typically are sent as continuous flows on a best-effort basis. Reliability comes from built-in redundancy in human interpretation of the media. Vision is quite robust in the presence of missing picture elements, while speech contains significant redundancy and in the worst case the listener can ask for repetition of what was said. An equivalent to this in DVS is the method used by DIS: all state data is retransmitted continuously, with gaps filled by dead reckoning. This method is extremely expensive in terms of network capacity, but it does provide an acceptable level of reliability when adequate network capacity is available.

Negative Acknowledgement (NACK)-based reliability: Any practical network will deliver successfully many more packets than it drops, so the first step toward RM is to use a negative acknowledgement when data is not received, rather than a positive acknowledgement when data is received. This approach is called "receiver oriented" because its success is centered on the receiver sending the NACK. It has the drawback that the sender cannot be sure the messages are received.

NACK suppression: While NACKs happen less frequently than ACKs, they still pose the problem that they may come in large batches if many receivers sense a lost message at nearly the same time. A workable solution to such "NACK storms" is to have the receiver set a timer for a small, randomly-selected value when it senses a lost message, and listen for NACKs until the timer expires. If a NACK pertaining to the lost message is received, some other receiver's timer has expired first, thus no further NACK needs to be sent. If not, the receiver sends the NACK to the multicast group.

Heartbeats: Another problem with use of NACKs is that, unless messages are sent at regular intervals, the receiver has no way of knowing it has missed a message until a subsequent message is received. A solution to this problem is to send a periodic "heartbeat" message at a low rate. The heartbeat bears the SN of the last message sent, and is used to trigger a NACK where appropriate. The heartbeat can be continuous, or it can take place several times following the message and then stop. It also can be omitted when another data message follows the first in close sequence.

Retransmission: The requirement for RM implies that at some point the lost data must be transmitted again. In DVS the missing data causes delays in accurate real-time simulation, so retransmission will happen as soon as the sender is informed it is needed. However, other RM applications may work in different ways. For example, another major application for RM is one-to-many distribution of files, such as might be done with a terrain database before a DVS session begins. In multicast file transfer it is possible for the NACKed messages to be retransmitted at the end of an initial, complete transmission of the file.

Loggers: To avoid the need to go back to the sender for a repair, a computer nearer to the receiver can act as "logger" and send a repair in response to the NACK if it has a copy of the message. [6] describes a system that uses logging, possibly combined with heartbeats. If the logger does not have a copy of the data needed for the repair, it sends a NACK to the source of the missing message.

Hierarchy of loggers: The logging process can be scaled to larger dimension by creating a hierarchy of loggers in virtual tree structure, distributed across the network. In this way, the repair arrives sooner because it comes from a logger closer to the receiver. Also the number of NACKs and resulting network loading is reduced in the average case, assuming losses are evenly distributed. [15] describes the "Reliable Multicast Transport Protocol" (RMTP) which uses such a system of loggers. A limitation of RMTP is that the hierarchy of loggers must be manually configured before the system is used. This is useful for relatively static groups such as might exist in business systems, but it would be problematic in a large, dynamic multicast group for DVS.

Forward error correction (FEC): A promising approach that is likely to be especially useful where round-trip times are long, or no repair request is possible (as in satellite-based multicast), involves sending sufficient redundant such that the receiver can reassemble the transmitted information without any retransmission [10]. Correction of errors within received packets is possible, using techniques such as the well-known Hamming code. However, a more powerful approach is to transmit information over successive packets such that the contents of packet that is completely lost can be recovered. The latter approach necessarily generates significantly increased traffic, but is necessary where FEC must provide repairs for packets dropped in congested networks.

2.2: Reliable Multicast for DVS

DVS has some characteristics that must be taken into account when developing an RM solution. Two characteristics of DVS that can be used to good effect in RM are:

Only the latest value of any object attribute variable is required by the simulation. This is important because it means the RM technique need not deliver, in order, the intervening values as TCP would do. Instead the message with the latest SN can be logged and delivered to the application at the receiver.

Some object attributes change rarely (e.g. appearance) while typically a few attributes change frequently and so are transmitted at regular intervals (e.g. position). While DIS sends all of this data in every protocol data unit (PDU), [3] points out that it is more efficient to send the rarely changing data reliably, and the frequently changing data as best-effort. If lost, the best-effort data will be replaced in by the next transmitted value and therefore will not need to be retransmitted. The amount of data transmitted reliably can be large and still require minimal network capacity, if it changes rarely. Further, in DVS the number of attributes requiring BE transmission typically is small. As a result, Cohen's approach can reduce network requirements for DVS considerably.

2.3: Optimal Combination of RM and BE Techniques for DVS

In [17] we built on Cohen's insight, introducing an important new idea: *A single transport protocol can achieve synergy by combining RM of rarely-changing data with BE transmission of frequently-changing data.*

The BE transmission is used like a heartbeat to carry the most recent SN, to trigger the NACK process for lost RM messages. Under our Selectively Reliable Transport Protocol (SRTP), retransmission is organized by multicast groups and simulation object classes (called "categories" in the original SRTP). In essence SRTP expands TCP's unicast address+port multiplexing structure to multicast group+class. Our current SRTP prototype implements this synergistic combination of RM and BE operation, and thus avoids incurring the additional transmission overhead that would be necessary with forward error correction.

In [18] we showed how to enhance SRTP using both NACK suppression and a dynamic, self-configuring system of loggers called Hop-Hierarchical Multicast Logging (HHML) to reduce network congestion. Under HHML, participating computers elect loggers at each level in a hierarchy of concentric circles within the network topology, based on the IP "time to live" (TTL) parameter. (TTL indicates the life of the packet as the count of "hops" between IP routers.) This results in (at least) a complete covering of the network with a hierarchy of loggers for each multicast group (see Figure 1). The loggers are organized by multicast group and class. They maintain the latest value of each object-attribute falling under RM. As with basic SRTP, when senders forward BE object attributes they append the latest SN of RM data for the object described by the message. The difference with HHML is that loggers distributed across the network keep copies of the latest value of each object-attribute that is covered by RM.

When a receiver sends a NACK under HHML, it causes RM data for the object to be retransmitted with TTL sufficient to reach the nearest logger. When the NACK is sent the receiver sets a timeout; if the timeout expires before the repair is made the NACK will be repeated. A new member joining the multicast group has the same effect as a NACK; the logger provides necessary data as an RM repair. In cases where the logger cannot provide a repair it sends a NACK to the next higher logger in the hierarchy, which then provides a repair to its whole TTL scope. This process continues up the TTL hierarchy until it reaches a logger that has the repair available. If necessary it goes all the way back to the source of the missing RM message.

We have begun work on a new prototype of SRTP which will include HHML. It also will include an optional, automatic heartbeat for classes with no BE messages. This will allow SRTP to provide a functional (but less efficient) RM where there is no BE DVS data to carry the SN needed for reliability.

A drawback of HHML versus statically configured routers is that the concentric circles are likely to overlap. Thus with HHML there is a tradeoff of some extra traffic in order to gain the benefits of dynamic self-configuration. The problem is worst with networks having long, thin multicast trees. In [16] we reported that the average redundancy lies between about 1.5 and 3.2 for a representative network that we have evaluated by simulation. As repairs are relatively infrequent activities (certainly less than one in twenty messages in an effective system), this means the extra traffic load from HHML can be expected to be at worst on the order of ten to twenty percent. We present below an idea for congestion control whereby SRTP could guarantee, in a network where other protocols cooperate to provide congestion control (as TCP does), that very few packets are lost due to queue overflow. This would mean that repair traffic would arise almost entirely from packets lost to transmission errors, in which case the network load due to HHML retransmissions for repair would be negligible.

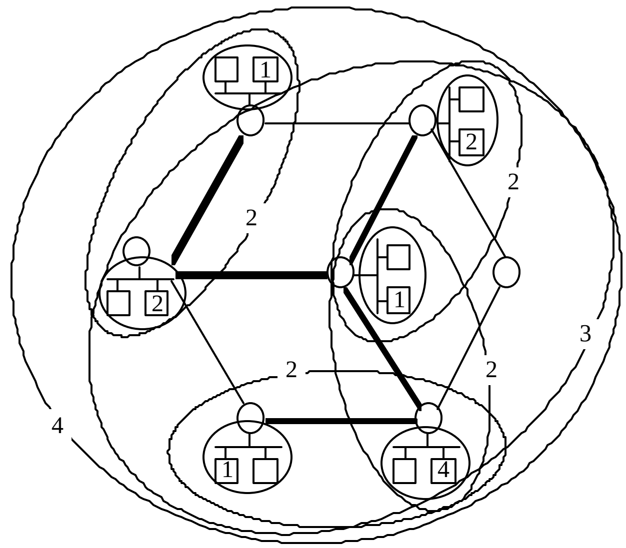

Fig.1- HHML Logger Hierarchy for a Multicast Group (numbered hosts are loggers; dark links represent the multicast tree)

3: Reliable Multicast For Distributed Virtual Simulation

We have investigated using SRTP to provide reliable multicast for two important forms of DVS that are used by the U.S. Department of Defense: DIS, and HLA used for real-time (STOW-like) DVS. SRTP provides three types of service, selected through the API by *mode* (see Figure 2):

- Mode 0 (best effort transmission) sends a best-effort message for the indicated class, including the SN of its last reliable transmission, to the multicast group (a special case, mode 0 class 0, goes to all SRTP subscribers in the multicast group).
- Mode 1 (reliable transmission) sends a reliable message for the indicated class, with a new SN value, to the multicast group.
- Mode 2 (lightweight one-to-one transaction) exchanges one reliable unicast datagram with another member of the multicast group, providing a light-weight, single-transaction form of reliability that is useful where the TCP connection process would unduly delay the real-time simulation.

The three modes are used within each multicast group to achieve a responsive, low-overhead transport that can meet the needs of distributed simulation efficiently in the large-scale multicast environment. Multicast groups can be selected to partition data to senders and receivers with shared interest criteria.

Figure 2 shows that SRTP can be implemented either directly over IP multicast, or (as in our prototype) over UDP so as to run as a user process. If implemented over UDP, the SRTP class can be mapped onto the UDP port so as to take advantage of the existence of the UDP port multiplexing.

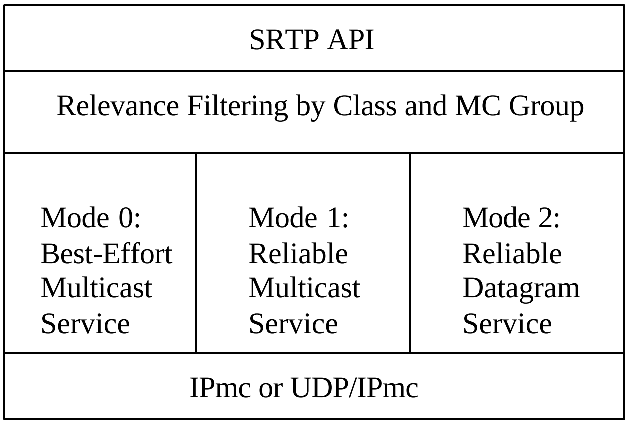

Fig. 2- SRTP Services

3.1: Reliable Multicast for DIS

The SRTP approach to RM for DIS was motivated directly by [3]. It uses a thin software adaptation applique (AA) at each DIS agent to map DIS into appropriate SRTP calls as follows.

(1) In the configuration process, data elements in the DIS PDU are divided into two categories: those expected to change rarely and those expected to change frequently.

(2) The AA compares each new DIS PDU from the agent to the previous one:

- Changes in frequently-changing elements are sent to the multicast group using SRTP BE (mode 0).
- Changes in the rarely-changing elements are sent to the multicast group using SRTP RM (mode 1). The applique could be structured either to send updates for one data element at a time, or to update the whole collection of rarely-changing elements together (because this happens infrequently, the impact on network traffic is minimal either way).
- Unicast transactions required by the DIS Collision PDU are exchanged using Mode 2.

(3) For incoming SRTP messages, the AA must maintain state received in mode 1 messages for all other participants in the group+class. When a mode 0 message is received, its contents are combined with this stored state to create a DIS PDU which then is sent to the DIS agent associated with the AA.

We have implemented the solution outlined above in the DIS-to-HLA converter used with our Light-Weight RTI, where it functions as expected. In [19] we show that this approach can result in better than 80% reduction in traffic for a typical DIS application.

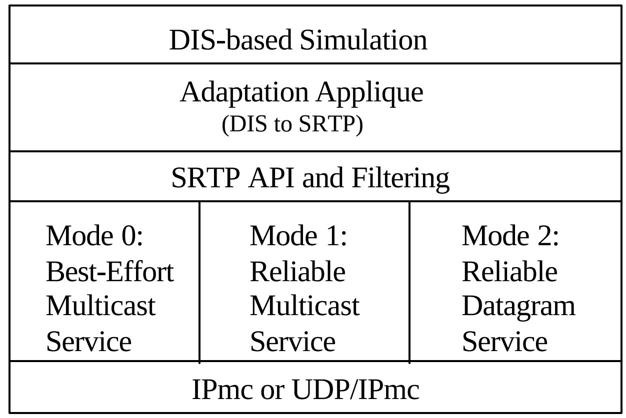

Fig. 3- Protocol Stack Supporting DIS with Applique and SRTP

3.2: Reliable Multicast for HLA

Our Light-Weight RTI was intended as an experimental tool to help us determine appropriate and productive functionality for RTI middleware, within the context of the HLA Interface Rules [4]. Since publication of [16] we have been investigating how SRTP might fit in the context of a practical RTI for DVS. We observed that there is a need, common to DVS systems, to transmit considerable amounts of state data infrequently and small amounts of state data at close intervals whenever a simulated object's state is changing. These are exactly the capabilities provided efficiently by SRTP. We therefore concluded that SRTP has the capability to provide the multicast communication services that were implemented by the TCP-based multicast emulation for STOW described in [22]. Moreover an SRTP-based RTI will be scalable to many computers and sites, unlike the STOW solution. In such an RTI, SRTP would function as a standardized building block to provide reliable multicast communication. All of the principal RTI functions can be supported by SRTP services (see Figure 4). In the following we assume many-to-many multicast communication for all information exchange among HLA federates.

SRTP provides support for the types of service expected by the HLA RTI as follows:

- RTI "best effort" service is supported by SRTP mode 0.

- RTI "minimum rate" service is supported by SRTP mode 0 with an automatic heartbeat.
- RTI "state consistent" service of slowly changing attributes is supported by SRTP mode 1, with an associated mode 0 stream that can be used to carry an SN to detect lost messages.
- RTI "reliable" service cannot be implemented dynamically in real time by any approach we know; however non-real-time RM is available commercially as described in [14].
- SRTP mode 2 has no role in the RTI, because the HLA does not identify a need for point-to-point reliable transactions.

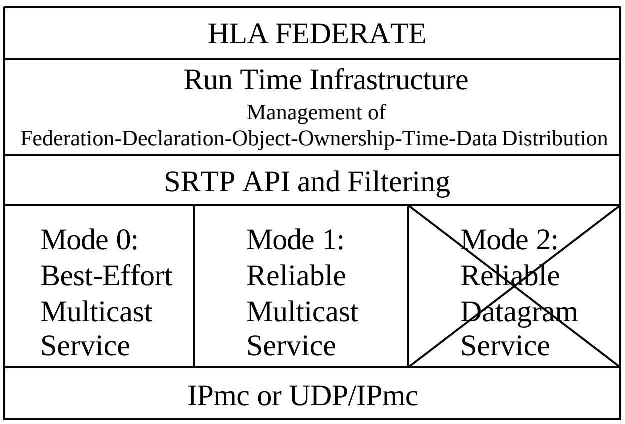

Fig. 4- Protocol Stack Supporting HLA with RTI and SRTP

For best effort, minimum rate, and state consistent communications SRTP provides class-based filtering, equivalent to TCP and UDP multiplexing to "ports." SRTP does not take the place of the RTI, rather it supports the RTI with RM. The RTI must continue to manage attribute ownership and distribution, and organize multicast groups. We have identified one specific change to SRTP that will be needed to support the HLA: the protocol fields for class-object (originally category-entity) must be extended to class-object-attribute, which will require expanding the associated header by two bytes to allow a sufficient address space. Other than this we believe SRTP, as we have proposed it, can provide improved network support for the HLA. As HLA facilities for employing data transport continue to grow more sophisticated, RTI developers should be able to use the hybrid RM/BE services such as those provided by SRTP to enable more effective DVS data transport over large multicast networks.

3.3: Future Work for SRTP

Our short-term plans for SRTP include:

Congestion Detection: One of the most important features of TCP is its ability to sense network congestion by increases in the round-trip time (RTT) between transmission of a segment and receipt of its ACK. We believe the mechanisms of SRTP will be easy to extend to a comparable function for multicast. Increase in time between transmission and receipt is an indicator of network congestion. By adding a time stamp to the SRTP message, it is possible to detect increased queueing delay in the paths from various senders on a one-way basis. One-way measurement must take the place of the RTT used by TCP because the round-trip measurement requires use of an ACK, which is impractical in multicast. The ability to detect increases in delay is independent of network-wide synchronization, because the parameter of interest is the change in the time stamp value, not its absolute value. However, if clocks are synchronized across the network using Network Time Protocol (NTP) [11] or Global Positioning System (GPS), it is possible to develop an reasonably accurate estimate (with NTP) or very accurate estimate (with GPS) of the true elapsed time in transit.

Congestion Control: TCP reacts to indications of congestion by adapting its sending rate downward. SRTP will be modified to achieve the same result. When congestion in the network path from some source is detected, SRTP will send a "flow control" (FC) message which would have properties similar to its NACK, i.e. suppression and HHML would be used. The FC message will go from all receivers sensing congestion to each sender (but with suppression to avoid duplicates), to inform the sender that congestion is being experienced. If the congestion is widespread or severe, the sender will respond by dropping some mode 0 (BE) messages before they are sent. The DVS system is designed to function without reliable delivery of these messages, therefore the rate of BE transmission can, if necessary, be reduced to the acceptable minimum of the dead-reckoning algorithm being used. This is greatly preferable to the alternative, a congested network that does not meet real-time requirements. If the SRTP sender is presenting traffic on the congested group for more than one class-object-attribute, it will drop traffic uniformly across all of them. As receivers detect that congestion has been reduced, another FC message will inform the sender that it is now possible to increase the BE traffic.

We note that, as in TCP, in SRTP the inference of congestion can be based on responsiveness of repairs. Moreover, this approach is scalable. Using HHML, a single lost packet for which the receiver finds a nearby repair need not trigger the congestion avoidance mechanism. Experimentation will be required to determine appropriate flow control algorithms, however a good starting point will be the TCP congestion control algorithm of [9] that provides aggressive sending rate reduction followed by slow rate increase. Work ongoing in the Internet Engineering Task Force (IETF) and Internet Research Task Force (IRTF) [MRBP98] is expected to result in algorithms that have similar properties in multicast networks where many senders and receivers must be considered. We expect these algorithms will be appropriate for SRTP congestion control.

Message Bundling: Because the largest part of messages under SRTP will represent rapidly-changing attributes, most messages can be expected to have a small amount of data and therefore a high ratio of overhead (header) to payload (data). A strategy used in STOW to increase efficiency of network use was to bundle together several messages from a source that are addressed to the same multicast group. This not only improves the ratio of payload to headers, it also reduces the absolute number of packets transmitted which lightens the workload of network routers because routing performance is a function of packets per second rather than bits per second. Bundling was planned in the initial SRTP design [17] but has not been implemented. SRTP will be modified to collect all messages within some relatively small time increment and bundle their transmission.

When we complete the new SRTP prototype with congestion detection/control and message bundling, we hope to demonstrate its use with both DIS and real-time HLA applications. Longer-term possibilities for SRTP are:

Investigate subcasting: The efficiency of HHML can be increased by restricting repair packets to the subnet which needs them, eliminating the redundancy inherent in the overlapping concentric TTL circles. Unfortunately this elegant approach requires a change to the operation of IPmc routers. We plan to use network simulation tools to investigate the potential improvement in SRTP from subcasting.

Investigate other applications: Other applications that combine small amounts of RM traffic with larger amounts of BE traffic should also be able to benefit from SRTP. At least one such application exists: distributed multimedia conferencing, which combines BE video and audio with reliable whiteboard graphics. We plan to investigate use of SRTP to support multimedia conferencing.

4: Summary And Conclusions

This paper has described the need for RM in distributed virtual simulation such as DIS and real-time HLA federations. The difficulty of providing a general solution for RM has been explained. Opportunities to take advantage of special characteristics of DVS have been identified, along with the way these characteristics have been exploited in the Selectively Reliable Transport Protocol (SRTP) to provide RM for DVS. Ways in which SRTP can support DIS and real-time HLA federations that will scale to large number of computers and objects have been identified. Some planned future improvements in the design of SRTP have been described. The approach described here can provide a practical, efficient solution to the RM problem for real-time DVS.

References

[1] Calvin, J., et. al., “STOW Realtime Information Transfer and Networking Architecture,” 12th DIS Workshop on Standards for the Interoperability Distributed Simulations, Orlando, FL, March 1995.

[2] Calvin, J. et. al., “Design, Implementation and Performance of the STOW RTI Prototype (RTI-s), Simulation Interoperability Workshop, Orlando Florida, March 1997

[3] Cohen, D., “Back to Basics,” *Proceedings of the 11th Workshop on Standards for Distributed Interactive Simulation*, 1994

[4] Defense Modeling and Simulation Office, *High Level Architecture Interface Specification 1.2*, U.S. Department of Defense, August 1997

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

[6] Holbrook, H., S. Singhal and D. Cheriton, “Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation.” *Proceedings of ACM SIGCOMM '95*, 1995

[7] IEEE 1278.1-1995, Standard for Distributed Interactive Simulation - Application Protocols

[8] IEEE 1278.2-1995, Standard for Distributed Interactive Simulation - Communication services and Profiles

[9] Jacobson, V., “Congestion Avoidance and Control,” *Proceedings of ACM SigCOMM ’88*, 1988

[10] Kermode, R., “Scoped Hybrid Automatic Repeat reQuest with Forward Error Correction,” *Proceedings of ACM SIGCOMM ’98*, 1998

[11] Mills, D. L., “Network Time Protocol (Version 3): Specification, Implementation, and Analysis,” RFC 1305, Internet Engineering Task Force, March 1992

[12] Mankin, A, A. Romanow, S. Brader and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," Internet Engineering Task Force RFC 2357, Internet Society, June 1998

[13] Macedonia, M., M. Zyda, D. Pratt, D., Brutzman, and P. Barham, "Exploiting Reality with Multicast Groups: A Network Architecture for Large-Scale Virtual Environments," *IEEE Computer Graphics and Applications*, September 1995

[14] Obrasczka, K., "Multicast Transport Protocols: A Survey and Taxonomy," IEEE Communications, January 1998

[15] Paul, S., K. Sabnani, J. Lin, and S. Battacharyya, "Reliable Multicast Transport Protocol (RMTP)," IEEE Journal on Selected Areas in Communications, April 1997

[16] Pullen, J. and N. Kakarlamudi, "Performance Issues for the Light-Weight RTI"," *Proceedings of the 3rd Simulation Interoperability Workshop*, 1998

[17] 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

[18] Pullen, J. and V. Laviano, "Adding Congestion Control To The Selectively Reliable Transmission Protocol For Large-Scale Distributed Simulation," *Proceedings of the Simulation Interoperability Workshop*, 1997

[19] Pullen, J., V. Laviano and M. Moreau, "Creating A Light-Weight RTI As An Evolution Of Dual-Mode Multicast Using Selectively Reliable Transmission," *Proceedings of the Simulation Interoperability Workshop*, 1997

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

[21] J. Pullen, M. Myjak and C. Bouwens, "Limitations of the Internet Protocol Suite for Distributed Simulation in the Large Multicast Environment," Internet Engineering Task Force RFC 2502, Internet Society, February 1999

[22] 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

Author Biography

J. Mark Pullen is an Associate Professor of Computer Science and a member of the C^3I Center at George Mason University, where he heads the Networking and Simulation Laboratory. He holds BSEE and MSEE degrees from West Virginia University, and the Doctor of Science in Computer Science from the George Washington University. He is a licensed Professional Engineer and a Fellow of the IEEE. Dr. Pullen teaches courses in computer networking, and has active research in networking for distributed virtual simulation and networked multimedia tools for distance education. Dr. Pullen received the IEEE's Harry Diamond Memorial Award in 1995 for his work in networking for distributed simulation.