Adding Congestion Control To The Selectively Reliable Transmission Protocol For Large-Scale Distributed Simulation

J. Mark Pullen and Vincent P. Laviano
Department of Computer Science and C^{3}I Center
George Mason University
{mpullen,vlaviano} @gmu.edu

Keywords:
HLA, multicast, selectively reliable transmission, congestion control

ABSTRACT: *We have developed a protocol to provide reliable transmission for the critical fraction of compressed DIS transmissions that provide reference data, by taking advantage of the stream of best-effort transmissions that provide transient data. This approach also has significant promise as a way to provide interoperable, re-usable components for the HLA RTI to be used for distributed virtual simulation.. In this paper we address issues of reliable multicast, with emphasis on the critical requirement for congestion control due to the fact that in a multicast network the large amount of control traffic arising from reliable transmission can overwhelm the data traffic in an "ACK implosion". We review the most promising techniques for congestion control currently used in reliable multicast research, and in particular a highly promising technique we call hop-hierarchical multicast logging that can greatly reduce control traffic while also providing very responsive repairs for lost reference data messages.*

1. Introduction

Large-scale distributed simulation, as envisioned by the High Level architecture (HLA), requires intercommunication among hundreds and potentially thousands of network hosts at many widely distributed network sites. In this paper we are concerned with distributed virtual simulation where simulated entities must communicate in real time on a many-to-many basis, with potentially thousands of entities participating in a single group. Under these circumstances it is essential to use *multicast* networking, where the underlying network is responsible to replicate messages flowing among the entities; the alternative that each entity sends copies of all messages to each other entity in its group is entirely impractical.

However, data networks are inherently unreliable in that they lose messages due to communications errors and queue overflow. Some distributed simulation approaches such as Distributed Interactive Simulation (DIS) deal with this problem by including the full state of the sender in every message. HLA, recognizing the great inefficiency of that approach, seeks to send only that information which changes. This however introduces a need for *reliable transmission* that is difficult or impossible to provide in a large, real-time multicast network. In this paper we show methods for achieving an acceptable approximation of reliable transmission for distributed simulation, with emphasis on the critical need for *congestion control* stemming from the fact that the acknowledgement messages (either positive or negative) needed for reliable transmission can represent an overwhelming amount of network traffic in a multicast network.

1.1 The large scale multicast environment

We assume here use of the Internet Protocol Suite (IPS) and in particular Internet Protocol multicast (IPmc). IPmc has been adopted by the distributed simulation community [IEEE95] because it has been shown to be scalable over a wide range of many-to-many multicasting applications. Based behavior demonstrated in the Internet multicast backbone (Mbone), potentially thousands of hosts representing tens of thousands of entities on hundreds of LANs using thousands of IP multicast groups can interact in an environment where multicast group membership is undergoing hundreds of changes per second. This operation may take place in a shared network with other simulations or networking applications, which in turn will require resource reservation to ensure that the simulations are ensured sufficient network capacity to maintain real-time operation [PMB97,SSM97].

1.2 Reliable transport in networking

Occasional message losses due to communications errors or temporarily overloaded switching queue buffers must be expected in any packet-switched networking environment. In unicast networking it has long been the practice to use a reliable transport protocol such as the Transmission Control Protocol (TCP) of the IPS which, when used over IP, can guarantee with a very high reliability that the required data has been transmitted, using an acknowledgement (ACK) or an implied negative acknowledgement (NAK) to the sender, in either case taking the form of the sequence number of the last correctly received byte. This unicast protocol will not work in a large-scale multicast environment because the large number of ACK/NAKs being returned to the sender would simply overwhelm the network, a process sometimes called "implosion". As a result, it has become the norm for multicast simulations to use the User Datagram Protocol (UDP) of the IPS, which provides only best-effort transmission and therefore does not use ACK/NAK.

1.3 Reliable transport and the HLA

Given the fact that the HLA is still undergoing refinement, it may seem premature to address its networking protocol requirements. However, in our area of interest (virtual simulation) the HLA must perform the same functions as DIS, a technology we believe we understand quite well. Moreover we have begun to investigate the functions of the HLA Run-Time Infrastructure (RTI) and have concluded that it must implement some form of reliable multicast transport. Therefore we are examining this problem in the contexts both of DIS and of the HLA RTI.

First DIS and now the HLA face a serious network problem in scaling up to large configurations. If the data is sent only when it changes, it must be sent reliably or anomalies will appear in the simulation output. However, even with the smaller amounts of information transmitted by the HLA, some messages critical to stable simulation output will be lost. For this reason Cohen has proposed [Coh94b] to break the data transmission into categories: *reference* values of attributes which are critical to the representation of an entity (for example, hatch open or closed), and *transient* values which are part of a continually refreshed stream of data, for example the position of the entity in a virtual simulation. (Cohen defines a third category, *absolute*, consisting of the collection of all reference values for an entity; this is a special instance of the reference case so we will not treat it separately here.) Loss of a transient value, while it may cause a small visual anomaly, will cause no lasting defect in the simulation. Given the critical need that reference values be conveyed reliably, we adopt Cohen's position and argue that it will be possible under the HLA to identify attributes that must be transmitted in reference mode, while other attributes may be treated as transient. We will show that it is possible to sustain reliable real-time transmission of reference attributes if they are a small fraction of the overall data stream, by treating transient and references attributes differently with regard to reliability, an approach we call *selectively reliable transmission*.

In the remainder of this paper we will assume that entity position, or at least the offset of position from some reference location, is a transient attribute. In that case, experience from DIS shows that a very large majority of the messages will in fact be transient, as the other state attributes change rarely and other sorts of interactions (such as collisions) represent a tiny fraction of overall network traffic [KeDo95]. We proceed with the understanding that under this condition not more than ten percent of all traffic in a multicast group is expected to require reliable transmission.

2. Selectively Reliable Transport

The function performed by TCP, UDP and our proposed Selectively Reliable Transmission Protocol (SRTP) is known in networking as a "transport", that is end-to-end connection between hosts. It occurs at a level directly above a network protocol such as IPmc.

2.1 The case for selectively reliable transport in distributed simulation

The unsuitability of TCP in a large-scale multicast environment has prompted most DIS application developers to use the User Datagram Protocol (UDP). UDP creates minimal overhead, meeting the latency requirements for DIS, but it provides no reliability. In many real-time applications, the loss or corruption of some small portion of an uncompressed data stream is tolerated because each new message supercedes all others, eliminating the value of retransmitting lost or corrupted data. This is the case with uncompressed DIS entity state information, but not with the reference

data when a reference/offset scheme is used to compress DIS entity state information, the offsets being treated as transient data. It is also not the case with reference data under the HLA.

Using UDP, each DIS application is forced to make its own provisions for reliable transmission of interaction and reference entity state data. This creates an unnecessary burden for application developers, and it violates the widely-accepted layering principles used to ease the design, implementation and maintenance of communications software of all types, including such application-layer protocols as the Real-Time Streaming Protocol (RTSP) and the Aggregate Level Simulation Protocol (ALSP).

The situation is somewhat better under the HLA, where the reliability of transmission is the responsibility of an RTI. However, the RTI must solve the same problem as the DIS application: reliable multicast of selected messages. In order to converge on interoperable RTI components it will be necessary to have a standardized protocol for the transport function. Therefore we contend that a protocol for selectively reliable transmission should be developed for use in the DIS and HLA multicast environment.

2.2 SRTP defined

SRTP provides three distinct transmission modes to meet the communication requirements for different classes of simulation data. Mode 0 provides a best-effort multicast service for transient data that does not require reliable transmission. Mode 1 provides a receiver-reliable multicast service for data that must be received reliably by all members of a multicast group. Mode 2 provides a reliable message service for data that must be received reliably with low latency by a single dynamically-determined member of a multicast group. Mode 2 uses unicast and therefore does not enter into considerations for reliable multicast and congestion control, therefore it will not be further considered in this paper. We do however note here that mode 2 has special value because the datagrams involved occur rarely, must be transmitted in real time, and may occur between and two arbitrary members of a multicast group, which makes use of TCP infeasible for a large group. Figure 1 shows the logical relationship of the services provided by SRTP.

SRTP supports multiple streams of entity state information by providing an entity ID field that is used to associate data belonging to the same entity state stream. The values contained in the entity ID field are unique per host, so the sending host's IP address (or some other unique identifier) can be prepended to this value to obtain a globally unique identifier for a stream. Reference/transient schemes for transmitting entity state information are supported by providing semantics that relate mode 1 and mode 0 data with the same entity ID, where reference data is transmitted using mode 1 and transient data is transmitted using mode 0.

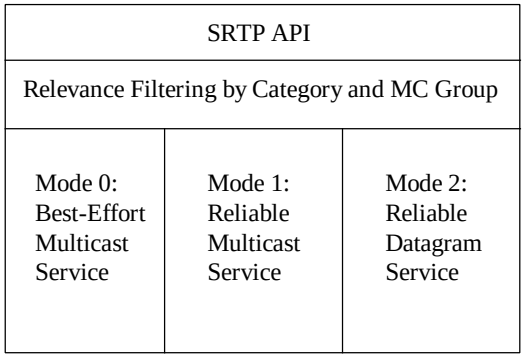

Figure 1: SRTP Services

SRTP provides for the notion of a category code, which is a label used to group related streams of entity state information. For example, the category code 3 might represent fixed wing aircraft. Client applications using SRTP are expected to be either DIS applications or components of an HLA RTI. These clients may choose to filter data on the basis of its category code, receiving only data of a particular category. Also, applications may choose to filter data on the basis of the multicast group to which it was sent.

3. Reliable Multicast Technology

Until recently, capabilities for reliable multicast have existed more in theoretical research than in practical development. However, reliable multicast capabilities now exist in products such as Multicast FTP [Star95] and wb, the Internet Mbone whiteboard [FJLMZ96]. Although most other reliable multicast approaches are still principally of academic interest, enough experience exists that an important portion of the multicast research community has reached an interesting conclusion: unlike the unicast case, it is unlikely that a “one size fits all” approach like TCP will apply to reliable multicast transport [FJLMZ96]. A fundamental reason for this is that requirements for reliability exist on a spectrum in each of several

dimensions. The portion of that design space we consider in this paper is the most demanding end of a continuum of latency requirements, needing reliable delivery of reference data in at most a few hundred milliseconds. On the other hand the absolute volume of data that might have to be supplied to make up for a lost message is quite small, to such extent that it makes sense to consider recovery logs scattered through the network to shorten the path to provide that data to the receivers that need it.

3.1 Issues in reliable multicast

There is a significant amount of ongoing research in the area of reliable multicast. Much of this work is focused in the area of one-to-many data distribution (e.g., audio, video, stock market updates), but there is also some ongoing work focused in the area of collaborative applications such as shared editors and whiteboards. While these applications are similar to distributed simulations in some ways, they generally take on a much lesser scale due to human factors. The following are some parameters that have been put forward in the reliable multicast community to categorize different classes of reliable multicast applications.

3.1.1 Receiver Reliability

A receiver-reliable approach is one in which the sender does not take explicit action (such as wait for a positive acknowledgement) to ensure that the data arrives intact at the receiver. Instead, the protocol provides mechanisms for receivers to detect the loss of data and to obtain lost data if desired. This approach has the benefit that the sender need not keep track of the set of receivers. In addition, the retrieval of lost data is strictly voluntary on the part of the receiver. Thus a receiver with a real-time constraint can avoid the latency of data recovery for non-essential data. The value of the receiver-reliable approach for reliable multicast has been described in [FJLMZ96, HCS95].

3.1.2 Number of Senders

In many reliable multicast applications, there is a single data source for each multicast group, and a potentially large group of receivers. By contrast, in distributed simulation multicast groups are associated with spatial, functional, or temporal classes of entities [MZPBB95] whose members may be spread across any number of hosts on a wide-area network. So, for each multicast group, there is not a single sender but rather many senders. Therefore, we categorize distributed simulation as a many-to-many application.

3.1.3 Real-time Requirement

In order for humans to meaningfully interact with distributed simulations there must be a strict upper bound on the end-to-end latency for the transmission of entity state information among simulators. For DIS, this bound has been found to vary between 100 and 300 milliseconds [DIS94], depending on the nature of the required interactions. A transmission protocol for distributed virtual simulations must be able to support these bounds.

3.1.4 Reliability Requirement

The real-time requirement for virtual distributed simulation fixes end-to-end latency to be below some bound. In other applications, where full reliability is required, the latency often varies as the state of the network changes. However, for virtual simulation, the level of reliability must vary as the state of the network changes, so that latency can remain fixed below the necessary bound. It is possible to take advantage of the abundance of transient data by varying the number of mode 0 messages sent to respond to changing network conditions, while sending all mode 1 and mode 2 data. Only under unacceptable network conditions would mode 1 or mode 2 data be dropped. The issue, therefore, is to maintain acceptable latency and overhead in providing the required reliability.

3.1.5 Ordering Requirement

The sequence numbering used by SRTP for loss detection provides ordering within each entity state stream. This is all that can be expected of a large-scale real-time simulation, as more robust ordering algorithms that might preserve overall ordering do not scale well to large groups.

3.2 Assumptions

Based on these insights we state several assumptions that we believe to be valid regarding the conditions under which SRTP must operate, so that we take advantage of these characteristics to improve the performance of SRTP.

- The application will be using reference/transient entity state compression to reduce network traffic. This is a reasonable assumption for large scale distributed virtual simulations, where it is needed

to constrain the cost of providing the required real-time network. SRTP mode 1 will be used for reference data, and mode 0 will be used for offset data. In other words, mode 0 data is that which changes rapidly and therefore is transmitted in a stream with sufficient frequency that it need not be transmitted reliably, while mode 1 data changes more slowly (or never) and therefore must be transmitted reliably.

- Each data stream originates from a single host at any given time. This fits the general case where the same host is responsible for simulating a given entity throughout a simulation exercise, and is true even where object attributes can be owned by simulations on different hosts and moved among hosts, in that each attribute data stream will still originate from a single host at any given time.

- A large portion (90 percent or more) of the multicast data generated by a distributed simulation is transient data that only requires mode 0 transmission, while some small fraction of the data is reference data that requires mode 1 transmission. We believe this must necessarily be true for distributed virtual simulation even under the HLA, because the most frequently changing data, location coordinates, must be updated at least whenever virtual position differs from dead reckoned position. (Non-smooth trajectories are characteristic of all significant combat behaviors, as predictable behavior leads quickly to death in combat. Other attributes, and their associated control information, necessarily change much more slowly, or the "fog of battle" would become utter chaos and not humanly comprehensible.)

- The application is likely to change multicast group registrations far more frequently than category registrations. This assumption derives from the expectation that categories will be used to represent broad classes of objects (e.g., tanks, planes, and infantry) while multicast groups will be associated with geographic areas of the virtual environment, which change as the entities move around the battlefield.

- Only the most recent mode 1 message of each entity state stream is useful. It is of no value to request retransmission of any but the most recent mode 1 data for a given entity ID and attribute, because this data would have been superseded by the new mode 1 data for the stream. This derives from the fact that simulators are only interested in information that best helps them to know the current state of each simulated entity. Therefore, the past states of such entities are irrelevant. (For HLA the most recent mode 1 message for each object-attribute must be retained.)

3.3 Congestion control mechanisms

Throughout the Internet, congestion control for unicast transmissions is achieved by the fact that the primary transport protocol, TCP, senses congestion and automatically reduces the rate at which traffic is presented to compensate. The multicast case is not so easy to deal with.

One of the primary goals of SRTP is to scale to extremely large distributed simulations, on the order of tens of thousands of simulated entities and hundreds sites distributed over a wide-area network. In order to achieve this goal with reliable multicast, one or more forms of congestion control are required. There are two principal sources of congestion in a distributed simulation: data traffic (both reference and transient) and control traffic (ACKs, NAKs, group setup, etc.) Congestion control for data traffic is needed in a both unicast and multicast environments, whereas congestion control for control traffic is much more needed in the multicast case. This is because, given a fixed loss rate for the entire network, the raw number of messages lost will increase as the number of receivers increases. Without some form of congestion control, each detection of these losses will result in a control message being sent by the receiver to the original sender, flooding the sender if the group of receivers is large. The following are possible approaches to congestion control for large multicast groups.

3.3.1 NAK Suppression

It is a well-understood fact that the use of unrestrained positive acknowledgements with a large multicast group of receivers will result in congestion, with each outgoing packet eliciting as many acknowledgements destined for the sender as there are receivers in the group. This is known as ACK implosion. A solution to this problem is to use a negative acknowledgement scheme instead. However, with a very large group of receivers or a very lossy network, even NAKs can still produce an implosion at the sender. A solution to this problem is to employ a NAK suppression mechanism, in which each NAK is delayed some random period of time in order to stagger the transmission of NAKs

back to the sender. In addition, a host that has scheduled a NAK for some time in the future must cancel the pending NAK if a NAK for the same data from some other host is heard while waiting. This approach has the drawback of requiring that NAKs be multicast to the entire group instead of unicast to the sender, but it is otherwise effective. NAK suppression was one of the congestion control mechanisms successfully employed during DARPA's STOW-E exercise [VHCS95].

3.3.2 Adaptive random backoff

The major challenge of adopting NAK suppression is determining the parameters of the random backoff interval. Ideally, these parameters should be a function of the round trip time between the sender and the receiver (because of the nature of multicast routing we can assume symmetric routes). While round trip time can be computed and used for protocol timers in the unicast case [Jaco88], it becomes more difficult in the multicast case. Because there are multiple receivers, there is no single round trip time. On the contrary, there are multiple round trip times that may vary significantly. Using a receiver-reliable approach in which the sender is not required to have knowledge of the individual receivers precludes any participation by the sender in calculating round trip times. The Scalable Reliable Multicast (SRM) approach used with wb has made successful use of adaptive random backoff [FJLMZ96].

3.3.3 Weighted averages

Another possibility is to use an average of round trip times, with recent values weighted more heavily than older ones. The utility of such an average is questionable, because it will underestimate the latency to far-away receivers. As an alternative it would be possible to use time-weighted average link latency to set the NAK suppression timer adaptively. One method of obtaining such a metric requires synchronized time among all participating hosts. This can be achieved using the Network Time Protocol (NTP) [Mills92], possibly supported by GPS. Using NTP, synchronized timestamps can be included in each message (or, alternatively, can be multicast to an entire group in a periodic "time heartbeat"). While this may seem to introduce significant overhead, it should be noted that the HLA requires that the RTI send timestamps in any case. Using the number of hops traversed and timestamp in a received packet, it is possible to arrive at an estimate of the delay per network hop, and these values can be averaged and weighted over time. Because the network for real-time simulation must meet strict latency requirements, under normal circumstances this value should remain stable. There does however remain a serious question when using a weighted average across the whole network, in that different portions of the network may have very different values of latency so that the average may represent some paths very poorly.

3.3.4 Local Recovery

In the case where data is lost at a significant distance from the sender, we believe it is useful to restrict NAKs to a local area in an effort to obtain a repair from a nearby host in order to avoid burdening the rest of the network with the request and repair traffic. Ideally, such an approach would keep traffic off of the WAN in cases where the data has been successfully received and buffered by a host on the LAN. One approach to implementing local recovery in conjunction with a NAK suppression mechanism is to scope multicast NAKs so that they travel no further than the number of hops necessary to reach the sender. It may also be desirable for the sender to scope repairs so that they travel no farther than the originator of the NAK.

3.3.5 Distributed Logging

Another approach to local recovery, described in [HCS95], is distributed logging. A logger is designated by the sender and, in addition to multicasting data to the group of receivers, the sender reliably transmits a copy of the data to the logger. The logger may be co-located with the sender. In addition to this primary logger, each receiver site has one or more secondary loggers. When a receiver detects data loss, a NAK is transmitted to the local logger. If possible, the logger responds with a repair. Otherwise, the secondary logger transmits a NAK to the primary logger, and the primary logger sends a repair to the secondary logger, which in turn is buffered and transmitted to the receiver who sent the original NAK. The logger may choose to multicast repairs, depending on the perceived number of losses. While it is desirable that each secondary logger support only a small to moderate number of receivers, this may not always be possible. In this case, NAK suppression can be introduced on the LAN to avoid flooding the secondary logger. If this approach is taken, it becomes necessary that NAKs and repairs be multicast on the LAN.

3.3.6 Hop-Hierarchical Multicast Logging (HHML)

[HCS95] suggests that it might be productive to use a multi-level hierarchy of loggers. With this inspiration we have begun to investigate hierarchical logging approaches at GMU. Our goal is to use hierarchy to localize NACKs for lost packets and resulting replacement messages. We believe hierarchy based on the functioning of the underlying IP multicast network is most likely to achieve this in an efficient, robust way. In this regard we have found very useful the feature of IP multicast called packet time to live (TTL) which places a strict limit on the number of hops a multicast packet survives. TTL can be used to localize NACK and replacement activity. Because each router decrements TTL by one when forwarding a multicast packet, TTL can also be used to determine how many hops a packet has come through the network. (Related work for the non-real-time environment has been undertaken by Paul et.al, [PSLB97], however their approach uses a static hierarchy.) The general approach in our design uses:

- loggers at every LAN for every multicast group active on any LAN host
- a hierarchy of LAN loggers that are responsible for circles of increasing radius along the path of the multicast tree for their group, with the first host to join the multicast group serving as the top-level in the hierarchy (see Figure 2)
- a discovery process where hosts newly joining a multicast group determine whether they must serve as logger, and receive initial states consisting of the most recent mode 1 segments that have been sent to the multicast group
- robust fail-safe procedures to ensure that a logger's place is taken if it dies, and also operate well even in the presence of errors in logger discovery and NAK

Our basic approach is for every SRTP host to perform logger discovery on each new multicast group it joins. If its LAN has no other logger for the group, it becomes the logger. It then probes by means of multicast packets with increasing TTL, and assumes the role of logger for circles of any TTL radius that does not have an established logger. Repeated for every joining host, this will result in full coverage of all hosts while NAKs and repairs will be localized to only those circles of hosts that have lost data.

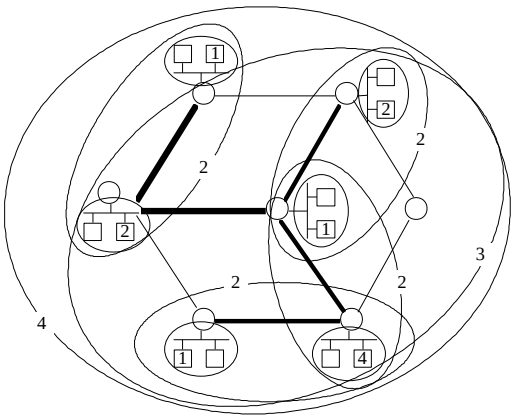

Figure 2: Hop-Hierarchical Multicast Logging
(numbered hosts are loggers; dark links represent multicast tree)

In any reasonably dense network the worst-case number of hops to repair a loss using HHML is in logarithmic proportion to the number of routers. A repair is guaranteed by the time the top-level logger in the hierarchy is reached. For example, in a symmetric network with each router connected to three others and having 96 routers, the distance from the furthest node to the top-level in the hierarchy is 5 hops in the optimal case, and 11 hops in the worst case. The average distance is half of this quantity, assuming uniformly distributed losses.

3.3.7 Adjusting the rate of mode 0 transmission to adapt to congestion

As described above, TCP reduces the transmission rate based on the level of congestion sensed in the network. Congestion is sensed as a function of the round trip time to the receiver and back. However in the multicast case we are faced with the problem of multiple round trip times in addition to a receiver-oriented environment where we do not want the sender to have to know about the receivers. It is possible to use an approach similar to that discussed above in the context of NAK suppression to estimate an average round trip time or average link latency. On reaching a designated threshold SRTP can begin to drop mode 0 data in favor of mode 1 and 2 data. Specifying a probability p, initialized to 0, that increases as a function of our congestion metric, it is possible to drop outgoing mode 0 messages with probability p. This would be combined with a TCP-like approach [Jaco88] in which p is increased by (1-p)/2 upon detecting congestion, and decreased back to 0 linearly each time our metric is safely below the congestion threshold. Such an aggressive slowdown combined with conservative return greatly reduces the risk of unstable

behavior that can be caused by multiple senders attempting to adapt to the same conditions.

4. Implementation Issues

4.1 Obtaining the TTL for Incoming IP Datagrams

In the operating systems we use it is necessary to be a privileged user to obtain most IP header information for an incoming datagram. A way to obtain the TTL value on a UNIX system using the sockets API is to use IP sockets directly to obtain the IP header along with incoming data. The obvious drawback to this approach is that we then have to replicate any desired UDP functionality. However, we believe the TTL information to be useful enough that we are considering this approach for future implementations.

4.2 Client Queue Management

The present implementation of SRTP queues data for client applications in order of arrival. A possible optimization is to have the protocol software remove from the queue any mode 0 or mode 1 data that has been superceded by a newly arriving message. This optimization should aid in preventing the client data queues from overflowing.

4.3 User Space versus Kernel Space Implementation

Currently SRTP is implemented in user space utilizing UDP and IP multicast. This allows us to take advantage of the UDP ports and checksum. However, because we are already considering bypassing UDP in favor of IP sockets in order to access IP header information, it is worth examining whether we should instead implement SRTP in the kernel. We expect that we would obtain a performance gain at the cost of implementation complexity and the loss of some flexibility in configuration.

4.4 Using UDP ports to filter categories

SRTP category codes are used to designate broad categories of entities. We have allocated one byte (256 values) for the category code, which we believe to be sufficient for this purpose. If SRTP is implemented over UDP, it is possible for UDP ports to filter messages. This would be achieved by creating a simple linear mapping between category codes and ports. (There are 65536 ports available, most of which do not have pre-assigned uses.) Here the overhead of binding and closing sockets is mitigated by the expectation that category registration and withdrawal is an infrequent event.

5. Conclusions and Future Work

Our investigation of a selectively reliable transmission protocol for distributed virtual simulation has yielded significant new insights from consideration of congestion control. Benefits of the reference/transient categorization of DIS or HLA data have become more clear. The potential reductions in latency for repair of lost reference data under HHML are particularly promising. Further, the potential of SRTP as a key component of interoperable RTI software appears to be great. We intend to proceed with implementation of congestion control in SRTP and use the resulting software to create a limited-function, distributed virtual simulation RTI for further experimentation.

6. References

[Coh94a] Cohen, D., "NG-DIS-PDU: The Next Generation of DIS-PDU (IEEE-P1278)", *Proceedings of the 10th Workshop on Standards Distributed Interactive Simulations, March 1994.*

[Coh94b] Cohen, D., “Back to Basics”, *Proceedings of the 11th Workshop on Standards for Distributed Interactive Simulation*, 1994

[DIS94] DIS Steering Committee, *The DIS Vision: A Map to the Future of Distributed Simulation*, May 1994.

[FJLMZ96] Floyd, S., V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing", *IEEE/ACM Transactions on Networking*, November 1996

[HCS95] Holbrook, H. W., S. K. Singhal and D. R. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation", *Proceedings of ACM SIGCOMM '95*, August 1995

[IEEE95] IEEE Standard 1278.2-1995, *Standard for Distributed Interactive Simulation - Communication services and Profiles*

[Jaco88] Jacobson, V., "Congestion Avoidance and Control", *Proceedings of ACM SIGCOMM 1988*, August 1988

[KeDo95] Kerr, R. and C. Dobosz, "Reduction of PDU Filtering Time via Multiple UDP Ports", 13th DIS Workshop on Standards for the Interoperability of Distributed Simulations, September 1995

[Mills92] Mills, D. L., "Network Time Protocol (Version 3): Specification, Implementation, and Analysis", RFC 1305, Internet Engineering Task Force, March 1992

[MZPBB95] 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

[PMB97] Pullen, J., M. Myjak and C. Bouwens, "Limitations of The Internet Protocol Suite for Distributed Simulation in the Large Multicast Environment", Simulation Interoperability Workshop, Orlando Florida, March 1997

[PSLB97] Paul, S., K. Sabnani, J. Lin, and S. Battacharyya, "Reliable Multicast Transport Protocol (RMTP)", IEEE Journal on Selected Areas in Communications, April 1997

[PuLa95] 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

[PuLa96] Pullen, J. and V. Laviano, "Prototyping the Selectively Reliable Transport Protocol", *14th Workshop on Standards for the Interoperability of Distributed Simulations*, March 1996.

[SSM97] Seidensticker, S., W. Smith, and M. Myjak, "Scenarios and Appropriate Protocols for Distributed Interactive Simulation", Internet Engineering Task Force Large Scale Multicast Applications Working Group, draft-ietf-lsma-scenarios-01.txt, work in progress

[Star95] StarBurst Communications Corporation, *StarBurst MFTP Compared to Today's File Transfer Protocols*, 1995

[VHCS95] Van Hook, D., J. Calvin, and J. Smith, "Data Consistency Mechanisms to Support Distributed Simulation", *13th Workshop on Standards for the Interoperability of Distributed Simulations*, September 1995.

Author's Biographies

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 recently received the IEEE's Harry Diamond Memorial Award for his work in networking for distributed simulation.

Vincent P. Laviano is a graduate student in Computer Science at George Mason University, and a Research Assistant in the C^3I Center Networking and Simulation Laboratory, where he has specialized in networking for distributed virtual simulation.