M.S. Computer Science Project

Rochester Institute of Technology

Rochester, New York

Framework for Throughput Analysis of Simple Reliable Multicast Protocol in a M2MP Network

By,

Romi Saluja

The M.S. Computer Science Project of

Romi Saluja is approved.

Faculty Chair: ___________________________________ **Date:** _____________

Faculty Reader: _________________________________ **Date:** _____________

Faculty Observer: ________________________________ **Date:** _____________

Acknowledgement

I would like to sincerely thank Professor Hans-Peter Bischof for accepting me as his graduate student, advising me through the course of this project and for ensuring my successful completion of the project.

I would like to sincerely thank Professor Alan Kaminsky for being on my Project committee and providing feedback and suggestions for the project report.

Table of Contents

List of Figures

1.0 Introduction

1.0 Anhinga Project

The Anhinga Project is a research effort being conducted at the Rochester Institute of Technology's Department of Computer Science. The objective of the Anhinga project is to develop an infrastructure to support collaborative applications such as chat, multiplayer games, etc. for running on an ad-hoc wireless network of mobile devices. The Anhinga infrastructure comprises of the Many-to-Many Invocation (M2MI) and the Many-to-Many Protocol (M2MP) libraries. M2MI is an application level API that makes available a new approach to developing collaborative applications. M2MI is based on object oriented design methodology and has many advantages over existing design methodologies such as no central servers, ease of application development and maintenance and is suitable for an ad-hoc wireless network. M2MP is a new broadcast messaging protocol that has been developed to support M2MI. M2MI uses the M2MP to send and receive messages in a wireless network of mobile devices with no device addresses [5][6][7].

1.1 Overview of this Study

M2MP has been developed to support M2MI applications in a mostly reliable wireless network. Currently, most M2MI/M2MP applications have been developed and tested in a reliable wired network [6][7]. There has been a requirement to test the M2MP in a real world environment i.e. an unreliable proximal ad-hoc wireless network of mobile devices, where devices are constantly joining and leaving the network. Also, M2MP provides unreliable multicast service. There is a requirement to build and integrate a reliable multicast layer to the M2MP implementation to provide reliable multicast service. This Master's project aims to satisfy the requirements by building a framework to test a reliable M2MP implementation in a simulated unreliable network.

2.0 Project Objectives

The objectives of this project are summarized as follows

?? A simple Lightweight Sender-based Reliable Multicast Protocol (LSMP) API will be designed, developed and implemented. The LSMP service will be built on top of the existing M2M protocol stack. The LSMP API will provide reliable service to applications running on

a M2MP network. The LSMP API will support many-to-many communication between senders and receivers. In addition, an approach to implement a Receiver-based reliable multicast protocol will be presented.

?? A M2MP Drop Packet layer would be designed, developed and implemented. The Drop Packet layer would be integrated into the existing M2M protocol stack. The Drop Packet layer would be used to simulate an unreliable network. It would provide an ability to drop packets being sent from senders to receivers and vice versa.

?? The existing M2MP throughput layer would be modified and integrated into the M2M protocol stack. The throughput layer provides an ability to control the send and receive rate of data packets by senders and receivers.

?? The existing Remote Control application will be modified to provide an ability to set the drop packet rate for the data packets using a GUI interface.

?? Tests would be carried out to analyze the maximum throughput achieved in simulated reliable and unreliable network conditions. The test would be conducted for one-to-many and many-to-many senders and receivers.

?? Design considerations for the multicast protocols will be reviewed and presented. Implementation of some of the standard existing multicast protocols will be discussed.

3.0 Literature Review

3.0 Introduction

This section provides an overview of some of features of a multicast protocol. The design consideration for each of the features is presented. In addition, design of some of the standard existing multicast protocols are discussed.

3.1 Multicast Protocol Design Considerations

A design approach to Multicast protocol development could be, to develop a generic protocol, which would be able to support a large number of varied applications. Another design approach would be to develop a protocol with limited set of features, suitable for a given type of application. The deciding factor for a given approach is ease of development and the type of applications supported.

The following is an overview of the features of a multicast protocol. In addition, existing protocols such as Log Based Receiver-Reliable Multicast (LBRM), Tree-Based Reliable Multicast (TRAM) and Scalable Reliable Multicast (SRM) are reviewed for each of the features.

3.1.1 Reliable

Reliable service in a multicast protocol can be either sender-based or receiver-based. In a basic sender-based reliability scheme, the sender is responsible for the providing the reliability service. The sender transmits data packets and the receivers send an ACK for each of the data packet. The sender maintains state information for all the receivers and tracks the receivers that have successfully received the data packets. If any of the receivers does not send an ACK in a given amount of time, the sender retransmits the data packets to all the receivers.

In a basic receiver-based reliability scheme, the receiver is responsible for providing the reliability service. The receiver detects lost or missing packets and notifies the sender. The sender then re-transmits the packets to the receiver.

Log Based Receiver-Reliable Multicast (LBRM) protocol implements receiver-based reliability using logging servers [4]. In this approach, the sender transmits data packets to a logging server and to the multicast group. The transmission of the data packets to the logging server from the sender is reliable i.e. the sender does not delete the data packets from its cache until it receives an ACK from the logging server. When the receiver determines that it has lost some packets, it requests the packets from the logging server instead of the sender. The logging server then retransmits the packets to the receiver.

Tree-based Reliable Multicast (TRAM) protocol also implements receiver based reliability [1]. TRAM organizes the sender and receivers in a hierarchical tree organization with the sender as the root of the tree. The sender forms a repair group by accepting receivers as members of the group. The receivers in turn form additional repair groups at the next level in the hierarchy by accepting other receivers as members. The receivers that form repair groups are designated as group heads for that group. The process continues until all receivers are part of a repair group. Data is cached at the sender and all the repair group heads. TRAM supports both, a top-down or a bottoms-up approach to

the tree formation. In the top-down approach the receiver has to be part of a tree before it can form its own repair group. In the bottoms-up approach, the receiver can form a repair group before joining the tree.

The sender transmits packets to all the receivers in its group. The receivers in turn cache the data and re-transmit the packets to all the members of their groups. Each member of the repair group sends an ACK to the group head for every data packet received. The group head in turn send an ACK up the chain, all the way up to the sender. The acknowledgements sent by the members of the group to the group head contain a bitmap of received and missing packets [1]. The group head retransmits the missing packets from its cache to the members and purges the packets, which have been successfully delivered, from its cache.

The Scalable Reliable Multicast (SRM) protocol is suitable for shared distributed applications [2]. In SRM all the participants are senders and receivers i.e. all participants send and receive data unlike the LBRM and TRAM protocols. Having said that, SRM also implements receiver-based reliability. The participant that receives data detects lost data and initiates recovery. When a receiver detects a lost packet, it sends a request for a repair packet to all the participants in the session. This is unlike the case in LBRM and TRAM where an acknowledgement is sent only to the sender or to a repair head. Also, one of the assumptions in SRM is that all data is persistently named. Thus the receiver requests the repair packets by its name instead of indicating missing sequence numbers. The request for the repair packet can be serviced by any other participant, which has the data. SRM implements optimization mechanisms to prevent duplicate repair requests and repair responses.

3.1.2 Scalable

In unicast transport protocols such as TCP/IP, there is a dedicated connection established between the sender and the receiver. The receiver sends control information to the sender indicating successful or unsuccessful transmission of each packet i.e. ACKS or NACKS. In a multicast setup where the number of receivers could scale up to thousands, the sending of control information (ACKS or NACKS) to the sender could result in the Implosion Problem i.e. flooding of the sender with control packets from the receivers. To avoid the Implosion Problem and make the multicast protocol scalable, the amount of control information sent to the sender needs to be reduced.

LBRM implements a “Statistical Acknowledgement” scheme to make the protocol scalable [4]. The idea is for the sender to statistically select a control group of secondary servers to act as Designated Ackers as opposed to receiving control information from all the receivers. If the sender receives an ACK from all the secondary servers in the Designated Ackers group, it does not multicast the retransmission of a given data packet. Then, if any individual receivers require the data packet to be retransmitted, the receivers would need to send a request directly to the sender. The sender would then unicast the packet directly to the receiver. If the sender does not receive an ACK from all the secondary servers in the Designated Ackers control group, it determines that sufficient number of receivers have not received the data packets and multicasts the data packet. Statistical Acknowledgement helps in scalability of the protocol by preventing NACK implosion in certain cases [4].

TRAM avoids the implosion problem at the sender, by designating the group heads in the tree formation, as the repair heads. The repair heads provide localized repair service to the members in their group. The members request missing data from the group head instead of the sender. The implosion problem at the group heads is prevented by limiting the number of members that a group can have. Another advantage of repair groups is the proximity of the group members to the group head. Repair packets can be sent with a small time-to-live (TTL) value resulting in network bandwidth consumption efficiency and also ensuring that repair packets are not sent to receivers in other repair groups that do not need them [1]. TRAM also minimizes the control information generated by aggregating the information at each level in the hierarchy and propagating it up the chain.

In SRM, the implosion problem can be a request implosion i.e. when all participants in a session detect a lost packet and transmit a repair packet request or a response implosion i.e. when all participants in a session receive a repair request and re-transmit a repair packet. When a receiver in an SRM session detects a lost packet, it waits for a random time determined by a function (request function) and sends out a repair packet request [2]. This function is based on the distance of the receiver from the source of the data packet. When a participant receives a repair request, it waits for a random time determined by a function (response function) and transmits the repair packet to all the

receivers. This function is based on the distance of the participant sending the repair packet to the participant requesting the repair packet [2]. These request and response functions are optimized for different network topologies such as chain, star and tree. The goal of the optimization is to limit duplicate generation of repair requests and to reduce the recovery delay. Based on simulation studies of the request / repair algorithm in the SRM protocol for different network topologies done by [2], the protocol is scalable i.e. one request and one repair packet is generated on loss detection and the recovery delay is consistent for different session sizes, in a chain topology and tree topology where all nodes are participants in the multicast session but does scale well in a sparsely populated tree topology. In addition, SRM limits the amount of bandwidth that the control packets can consume to a small percentage of the total available bandwidth.

3.1.3 Loss Detection Mechanism

In a basic receiver-reliable protocol, the receiver detects lost or missing packets. The sender periodically sends data packets to the receivers. Each of the data packets references an incrementing sequence number. In addition to the sequence number, the sender also specifies a Maximum Idle Time (MaxIT) in each packet [4]. The sender guarantees the receivers that it will transmit a packet every MaxIT. In the case where no more application data is available to be sent, the sender transmits "heartbeat packets". These heartbeat packets contain the sequence number of the last data packet sent by the sender and the MaxIT value. The receiver determines that it is has not received all the data packets i.e. loss detection, if the sequence number in the data packet is not the next expected sequence number or if no data or heartbeat packet is received for MaxIT time.

LBRM optimizes on the above approach and implements a variable heartbeat scheme to detect loss packets in a multicast transmission. In a variable heartbeat scheme, the time interval to transmit heartbeat packets after a data packet transmission is bounded by a minimum and maximum time interval (h_{min}, h_{max}) [4]. The idea is to send a heartbeat packet h_{min} time after the data packet transmission and then continue to send the heartbeat packet at twice the current h_{min} value until the interval reaches the h_{max} value. At the point the heartbeat packet is sent at h_{max} interval. In a fixed heartbeat scheme, this time interval is set to a constant value (h_{max}). Based on performance analysis of a variable and fixed heartbeat scheme [4], the variable heartbeat scheme results in faster loss detection and lower number of transmitted heartbeat packets.

Senders in a TRAM protocol include a sequence number in very data packets. The receivers in the repair groups detect lost or missing packets when there is a gap in the sequence number of the received packets. In a TRAM protocol the sender does not transmit heartbeat or keep alive packets [1].

In an SRM application session, the data packets sent by all the participants include a unique host identifier and a locally generated sequence number [2]. The receivers detect a lost or missing packet when there is a gap in the sequence number from that particular host. In addition, every host periodically broadcasts session messages indicating the highest sequence number sent by that host so far. The receivers detects lost packets by looking at the sequence numbers in the session messages and comparing it to the sequence number of the data packets received so far from that particular host.

3.1.4 Retransmission Mechanism

LBRM uses distributed logging for retransmission of lost packets [4]. Each client site with multiple receivers has a secondary logging server. The receivers at the site request a lost packet from this secondary server, which in turn sends the request to the primary logging server. Using distributed logging for retransmission results in lower retransmission bandwidth as fewer retransmission requests are sent to the primary logging server. It also results in lower retransmission latency, as it would be faster to get a retransmitted packet from a secondary server at the local site than from a primary server.

In TRAM, the receivers send an acknowledgement to the group head for a given number of packets at a time. This number is configurable and is called the ACK window size. The acknowledgment contains the sequence number and a bitmap [1]. The bitmap indicates missing data packets if any, which triggers a retransmission from the group head. The retransmission scheme in TRAM is optimized to avoid duplicate retransmission of data packets.

In SRM protocol, the receivers do not send an ACK unlike LBRM and TRAM. On detection of lost packets, the receivers request the data by sending a repair request. Any host that has a copy of the requested data responds to the repair request by retransmitting the data [4].

3.1.5 Cache Strategy

The sender caches transmitted data packets. An important design consideration is at what point to purge the data from the sender's cache. Some of the cache strategies [3] are

?? Caching with Spatial Persistence – In this scheme a given number of most recent packets are cached.

?? Caching with Temporal Persistence – In this scheme, transmitted packets are cached only for a certain amount of time.

?? Caching with Logical Persistence – In this scheme, the amount of time packets are cached at the sender, is set dynamically.

In LBRM the sender purges any data from its cache, on receiving an ACK from the primary logging server. The amount of time the data is cached at the logging servers is application specific. Some applications might require data to be cached at the logging servers for a certain amount of time where as some other applications might require the data to be persistently stored by writing it to a disk.

In TRAM protocols, the packets are cached at the sender and repair heads. The data is purged on receiving an ACK from the members in the repair group indication the successful transmission of the packets. In the case of SRM, the caching strategy is application specific. The data can be cached persistently onto a disk or the most recent data is cached.

3.1.6 Packet Delivery Mechanism

Packet Delivery mechanism relates the order in which the packets were transmitted by the sender to the order in which the packets were received by the receiver. The different ordering strategies [3] are

?? Unordered – A multicast protocol implementing an Unordered packet delivery mechanism does not guarantee that the order in which packets are received at the receiver is the same order in which the packets were transmitted by the sender.

- ?? Source Ordered – A multicast protocol implementing a Source ordered packet delivery mechanism guarantees that the order of packets received at the receiver are in the same order as were transmitted by a particular sender.
- ?? Totally Ordered – A multicast protocol implementing a Totally Ordered packet delivery mechanism guarantees that the order of packets received at all receivers are in the same order as were transmitted by all senders.

LBRM does not implement any particular Packet Delivery mechanism. It is the responsibility of the application layer to implement a suitable delivery mechanism. Based on the fact that the he sender in TRAM protocol uses sequence numbers in the data packets, it would be reasonable to assume that TRAM implements the ordered packet delivery mechanism. SRM supports the unordered packet delivery model. It only guarantees eventual delivery of all the data to all the participants in the session [2]. If required by the application, a source or total ordered delivery mechanism can be built on top of the SRM protocol.

3.1.7 Late Join

Protocols supporting Late Join allow clients to connect to a multicast session after data transmission has already started and still be able to retrieve all the data that had been transmitted since the session began. In order to support Late Joins, there must be at least one host which has all the data cached since the session began [].

In TRAM protocol based system, Late Join clients are supported partially. Only partial previously sent data is available for recovery [1]. SRM by itself does not support Late Joins. But Late Join can be supported at the Application Level by pursuing a persistent data caching strategy at all the hosts in the SRM session. But this approach might lead to performance issues in the case of long-lived SRM session with a large number of participants.

3.1.8 Flow and Congestion Control

Multicast protocols should be able to detect network congestion and control the data transmission rate. TRAM implements flow control in two phases – slow start and congestion control [1]. In slow start phase, the initial data rate is a small percentage of the maximum data rate. The data rate is then increased gradually until it reaches the maximum data rate. The idea behind the slow start phase is to

test the network and detect network congestion. In the congestion control phase, congestion is detected at the receivers; repair heads or the sender and congestion reports are generated. Congestion is detected at the receivers, when the number of missing packets increases with every ACK window size of packets received. The repair head and sender detect congestion and generate congestion reports when the data cache grows beyond a set high watermark. On receiving the congestion reports, the sender decreases the data rate by a given percentage. In the absence of congestion reports, the sender gradually increases the data rate based on the historically highest achieved rate [1].

SRM does not have any flow and congestion control mechanisms. The argument is that SRM provides the basic scalable, reliable functionality that can be used for a large number of applications. As different applications have varying flow and congestion control requirements, this feature should be application specific and build on top of the reliable and scalable service.

3.1.9 Repair Scheduling

Repair scheduling determines the priority with which data packets, acknowledgements, retransmissions and control packets are transmitted over the network.
The possible approaches are [3].

?? Retransmissions are sent with higher priority than new data packets.
?? Retransmissions are sent with lower priority than new data packets.
?? Retransmissions and new data packets are sent on first-come-first-serve basis.

In the case of SRM, retransmissions for current data is given the highest priority, next comes new data packets and lowest priority is given to retransmissions for old data [2]. In TRAM, retransmissions are given higher priority over new data packets.

3.1.10 Network Topologies

Multicast protocols should adapt to varying network topologies. TRAM dynamically creates the hierarchical trees based on whether the nodes in the network are sparsely populated or densely populated. SRM supports chain, start and tree network topologies.

3.1.11 Types of Application

There are many different kind of applications for which a multicast protocol would be applicable. The applications can be one-to-many or many-to-many applications such as stock ticker or a chat application. In addition, some applications have a requirement for unordered or ordered delivery of data [1]. Some applications require fast detection of data loss [] while others require time bound delivery of data [1]. Given the varying requirements, it might not be possible to develop a generic multicast protocol suitable for all applications. A design consideration would be to determine the suitability of the different multicast protocols.

LBRM is suitable for applications, which require real time information and for applications, which require notification or dissemination of large amount of information such as Traffic Report and Stock Quote Dissemination [4]. TRAM is suitable for applications with a single sender and multiple receivers. They are designed to support bulk data transfers [1]. Some examples are File Transfer Protocol (FTP) applications, Stick Ticker program, etc. SRM is suitable for shared data applications with multiple senders and receivers such as whiteboard application.

4.0 Design

4.0 Architecture

The architecture of the current version of the M2M protocol is as shown in Figure 1. An application interfaces with the M2M protocol layer. The M2M protocol layer then uses a channel to send and receive messages from the underlying communication medium such as Ethernet. There are different channel implementations available such as IPMulticastChannel and IPUnicastChannel, to name two and any one of these different channel implementations can be used.

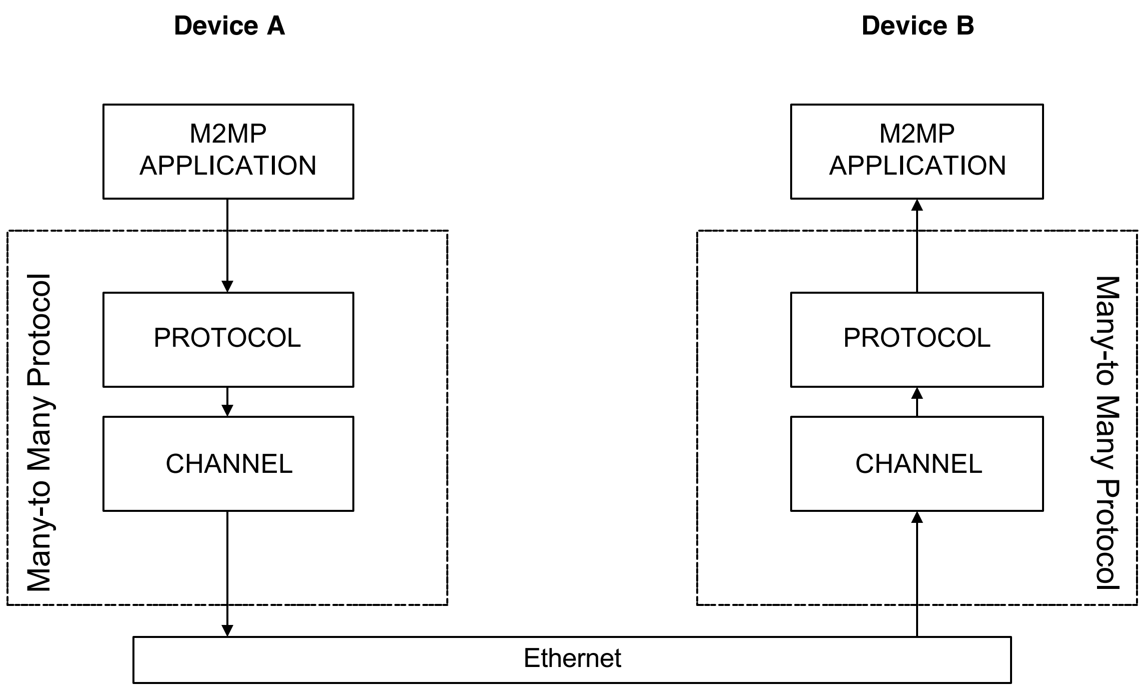


Figure 1: Current Architecture of M2M Protocol

As part of this work, the M2M protocol architecture will be modified as shown in Figure 2. The Drop Packet Layer and Throughput Layer will be integrated into the M2M protocol stack. The exiting M2M protocol internals will not be modified. The Drop Packet Layer will be implemented as an additional channel in the M2MP channel pipeline. In future, this would enable bypassing of the Drop Packet Layer in the M2MP channel pipeline, if required.

In addition, a LSMP API will be built on top of modified M2M protocol stack. This framework will provide the ability to replace the LSMP API with newer or additional reliable service in the future.

A M2MP application will be use the LSMP API to transmit and receive packets over the M2MP network. The Remote Control application will enable setting the drop packet rate and data packets send and receive delay to simulate an unreliable network.

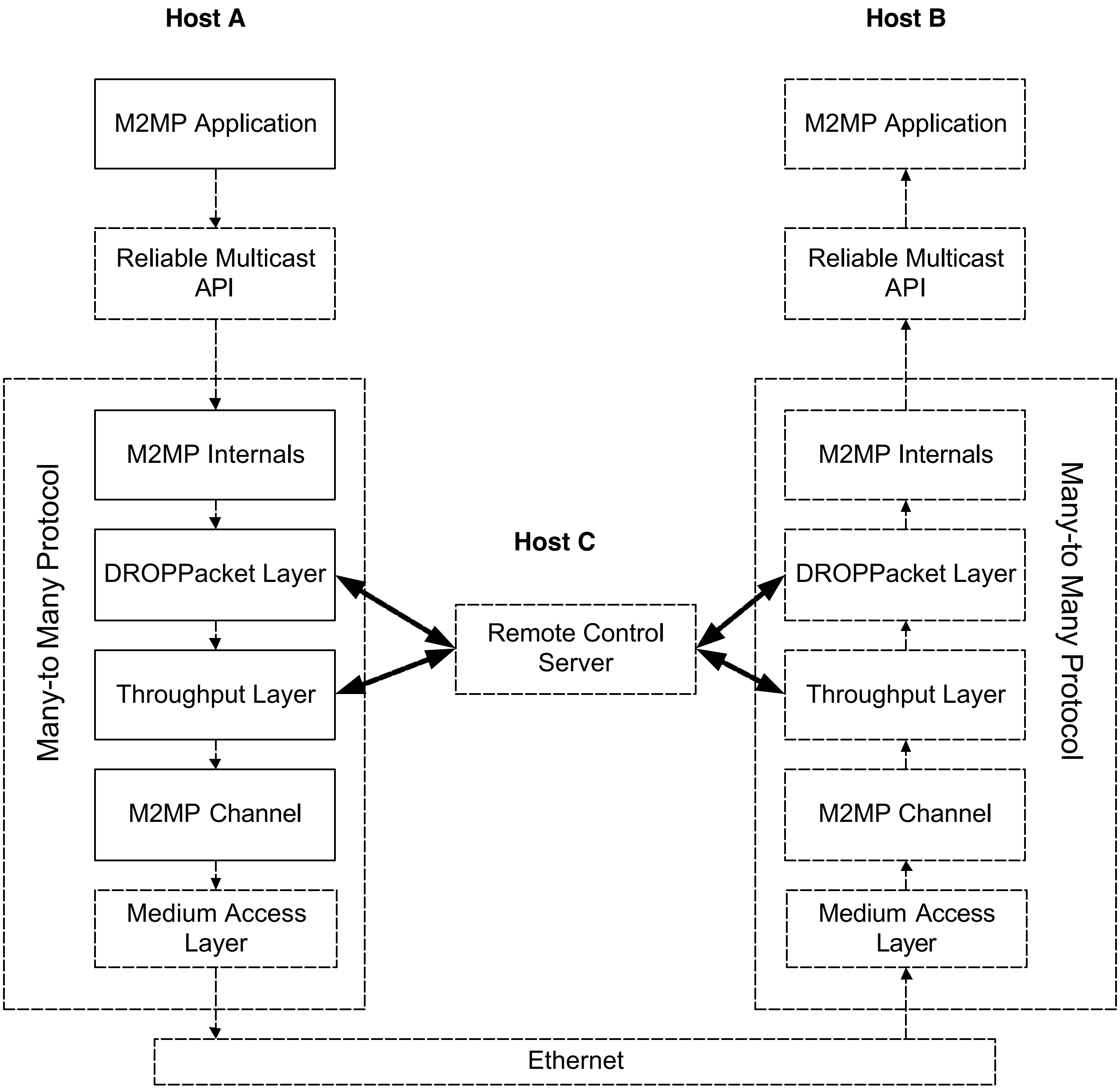


Figure 2: Proposed Architecture

4.1 Reliable Multicast API Layer

4.1.1 Approach

A reliable multicast layer will be built on top of the M2M protocol stack to provide reliable service to M2M applications as shown in Figure 3. The application will send application messages to the Reliable Multicast layer. The Reliable Multicast layer would then ensure that the application messages are reliable transmitted from the senders to the receivers.

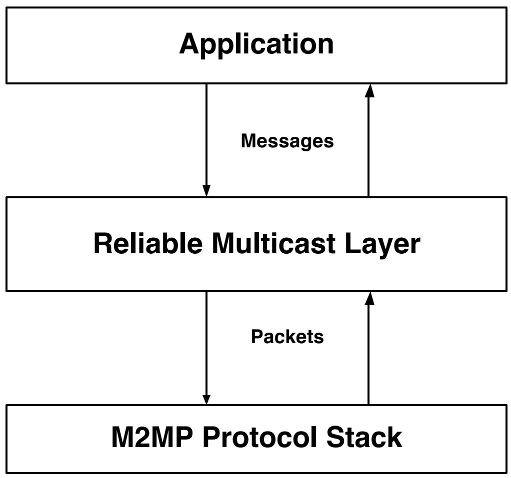


Figure 3: Reliable Multicast Layer

The Reliable Multicast Layer would have the following features

?? Support many-to-many communication

?? Lightweight

?? Implemented as an API

As discussed in Section 3.2.1, the reliable service in a multicast protocol can be sender-based or receiver-based.

4.2.1.1 Sender-Based Reliability

In the Sender-based reliability approach, the sender is responsible for keeping track of the receivers and ensuring that all receivers have successfully received the data packets. A simple Sender-based reliable approach is described as follows [11]. The sequence of events at the sender are described on the left. The sequence of events at the receiver are described on the right. The sequence of events take place in the given order from top to bottom.

Sender

Sender receives packets from the Application.

Sender buffers the packets.

Sender creates an ACKPending list of all Receivers

Sender multicasts the packets and waits for an ACK from all the receivers in ACKPending list.

Sender periodically keeps multicasting the packets until ACK is received from all the receivers.

On receiving ACK from a Receiver, the Receiver is removed from the ACKPending list.

When ACKPending = 0, Sender sends the next packet from the buffer.

Receiver

Receiver joins the session and registers with the Senders.

Receiver receives packet from the Sender. It checks for the Sequence number.

If no gap is detected, the packet is passed on to the Application and ACK is sent to the sender.

If a gap is detected, the packet is ignored and the ACK is sent for the last correctly received sequence number.

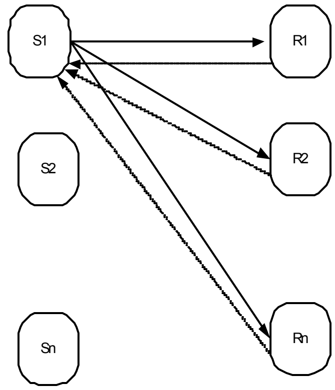


Figure 4: Sender-Oriented Protocol

4.2.1.2 Receiver-Based Reliability

In the Receiver-based reliability approach, the receiver is responsible for ensuring that all receivers have successfully received the data packets. A receiver may choose to retrieve the lost data packets or may ignore them. A simple Receiver-based reliable approach is described as follows [7]. The sequence of events at the sender are described on the left. The sequence of events at the receiver are described on the right. The sequence of events take place in the given order from top to bottom.

Sender	***Receiver***
Sender receives packets from the Application.	Receiver receives packets from the Sender.
Sender buffers packets. The number of packets buffered is equivalent to the Send Window size.	Receiver buffers packets. The number of packets buffered is equivalent to the Receive Window size.
Sender sends packets to Receiver at a given transmission rate R (1 packet / second)	If no gap is detected the packet is passed on the Application.
On receiving a NACK, the Sender	If a gap is detected a NACK is

calculates the difference between the current Sequence Number in transmission and the Sequence Number in the NACK packet. If the difference is greater than half the Send Window size, the transmission rate is reduced to 0.25R.	multicast to the Sender.
On receiving a NACK, the Sender multicasts the corresponding repair packet from the buffer.	Duplicate data packets are ignored
On receiving a NACK, a Timer is started. If no further NACKS are received during that time, the transmission rate is reset to R (1 packet / second).	
Duplicate NACK packets are ignored.	

4.1.2 Design Description

The sender-based reliability service Lightweight Sender-based Reliable Protocol (LSMP) was implemented as part of this work.

LBRM, TRAM and SRM are suitable for large number of multicast applications. But they would not be suitable for reliable multicasting in an ad hoc mobile network. The main characteristic of an ad hoc mobile network is high number of transient nodes i.e. nodes joining and leaving the network or moving from one multicast group to the other. The above-mentioned protocols would not be able to adapt to this highly dynamic network. [11] proposes a reliable multicast algorithm for ad hoc networks. As per this approach, a group of senders service the receiver nodes in a given service area. The sender transmits data packets to the receivers. The receivers in a given service send an

ACK for every packet received. The sender is aware of the all the receivers in its group and keeps retransmitting the data packets until it has received an ACK from all the receivers in its group. If the receiver moves from the service area of one sender to another sender, the old sender does a handoff of the receiver, to the new sender. This ensures that the receiver still is able to receive all the data packets. LSMP is a simplified version of the algorithm proposed by [11]. LSMP senders and receivers interact as per the approach described above. But LSMP does not implement group maintenance. At the start of the session, the senders are aware of the number of receivers participating in the session. This enables the senders to ensure that they have received ACK messages from all the receivers. But this approach has certain limitations. A receiver cannot late-join a session as the sender would not be aware of it. The receiver would receive the data packets from the sender but sender would not process any ACK packets from this receiver. But a sender could late-join a session and send packets and process ACK from receivers. Additionally, if a receiver participating in a session were to disengage or become unavailable, the senders would keep re-transmitting the same data packet, as the unavailable receiver would no longer be transmitting ACK packets. But if a participating sender were to become unavailable, the session would still be active, as the receivers would continue to receive data from the other senders.

LSMP has two design components – Reliable Send and Reliable Receive as described below.

4.1.2.1 Reliable Send

The design for reliably sending data from the senders to the receivers is as shown below in Figure 5. The M2MP application uses the LSMP API to pass the Application Messages to the Reliable layer.

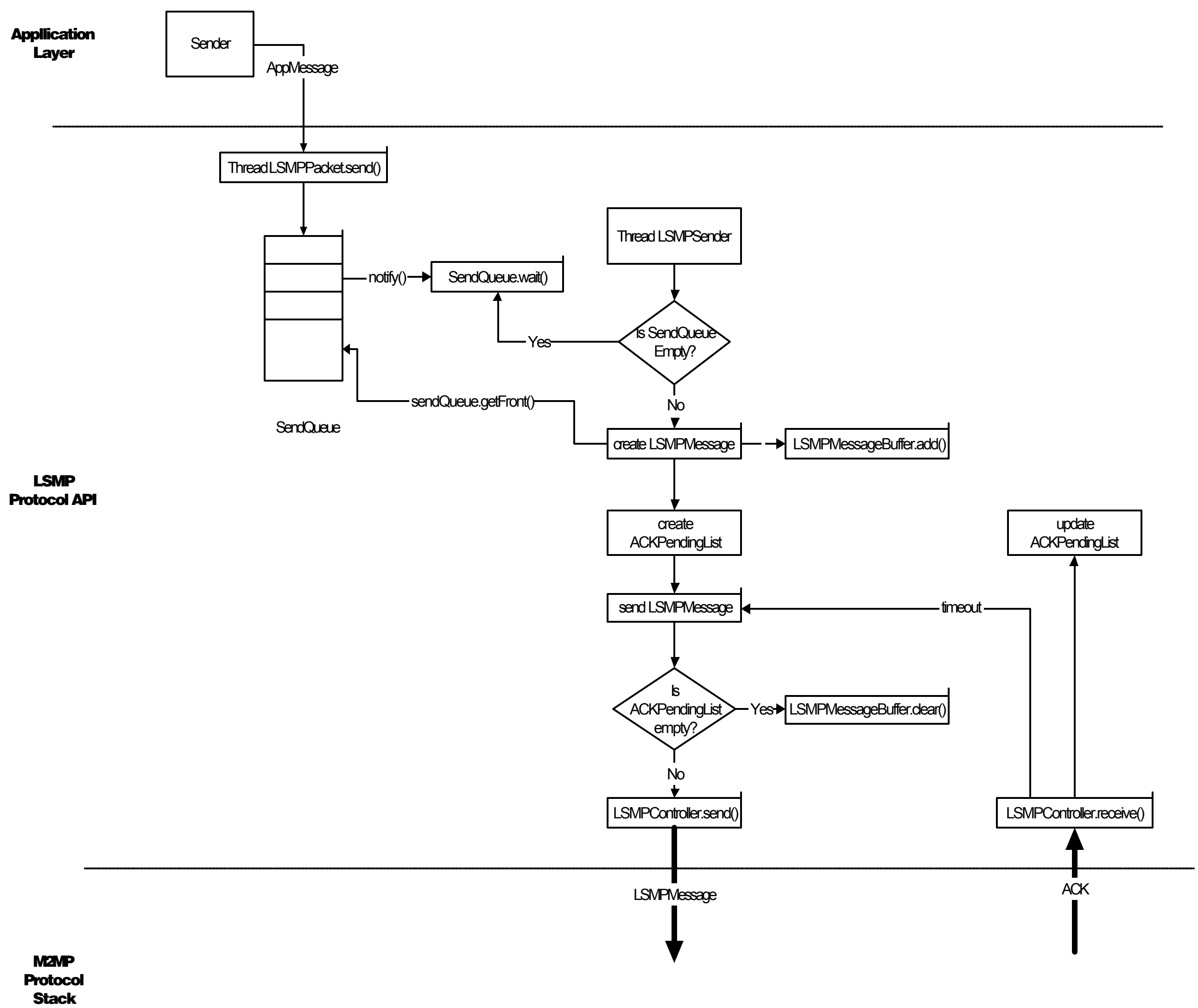


Figure 5: Reliable Send Design

The application messages are buffered in the LSMP send queue. A LSMP sender thread is synchronized on the send queue and is notified when application messages are added to the send queue. The sender thread retrieves the message from the front of the queue and encapsulates the application message into an LSMP message object. In addition, the sender thread creates a list, called the ACKPending list, of the number of receivers that are currently part of the multicast group and will be receiving the LSMP message object. The sender is provided the information on the number of receivers participating in the session at the start of the session. The LSMP message object is then passed to the M2MP layer, which transmits the message object over the network to the

receivers. The sender thread then waits to receive an acknowledgement (ACK) from all the receivers that are members of the multicast group.

On receiving an ACK from a receiver, the sender thread removes the receiver reference from the ACKPending list. When an ACK have been received from all the receivers, all receiver references are removed from the ACKPending list. When the ACKPending list is empty, the sender thread retrieves the next application message from the sender queue to pass on to the M2MP layer.

In addition to the primary sender thread, another asynchronous sender thread is started to periodically resend the message object until an ACK has been received from all the receivers.

4.1.2.2 Reliable Receive

The design for reliably receiving data is as shown below in Figure 6. A receiver thread in the LSMP layer constantly listens for incoming messages. If the sequence number in the incoming message is not the expected sequence number or if the message is a duplicate, the receiver thread ignores the message. If the message is valid and has the correct expected sequence number, the message is put into a receive queue. In addition, an ACK for the message is sent to the sender.

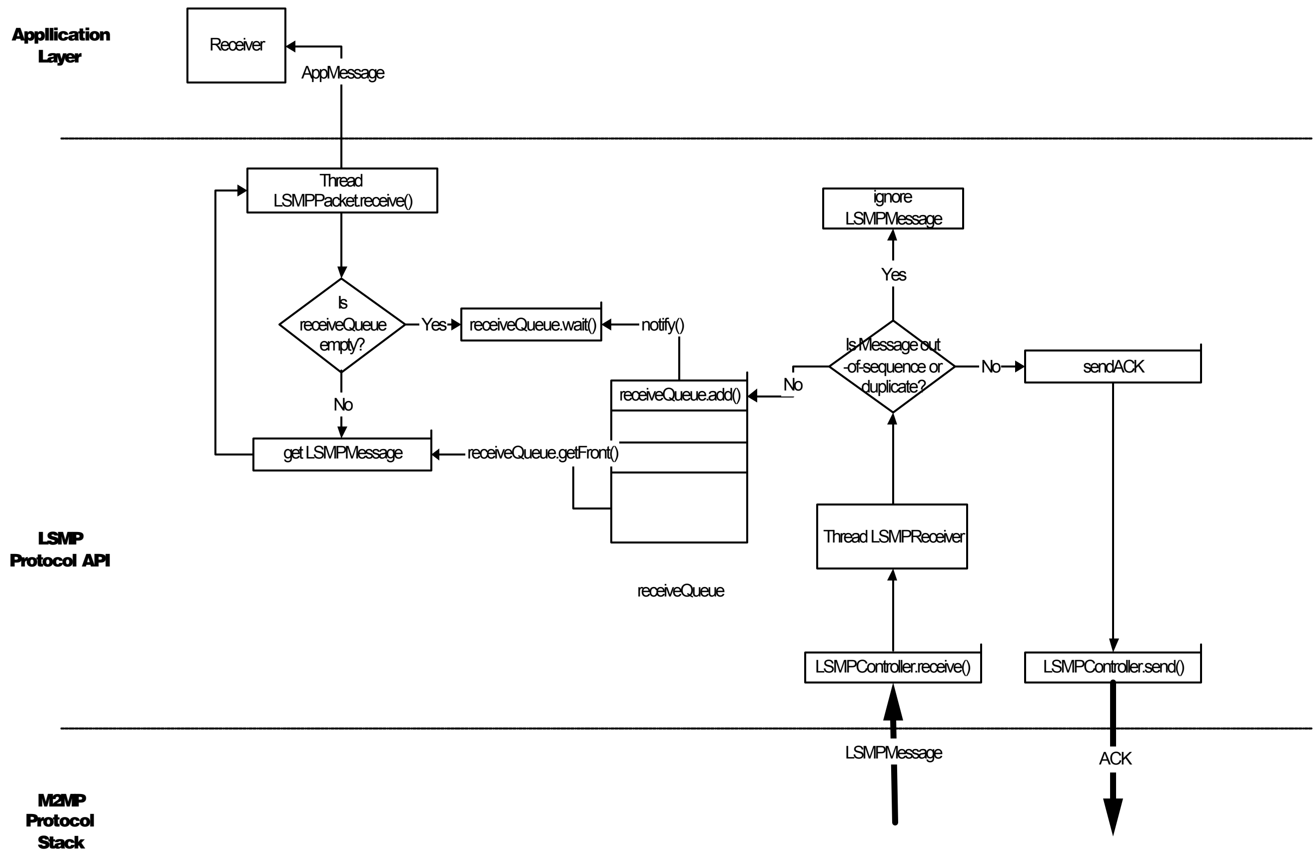


Figure 6: Reliable Receive Design

A different receiver thread is synchronized on the receive queue object and is notified when messages are added to the queue. The message is then retrieved from the receive queue and passed on to the Application.

4.1.3 Communication Sequence

The following is a brief description of all the classes that make up the LSMP API layer. LSMPPacket – The LSMPPacket class contains the send and receive methods that the M2MP application uses to send and receive messages. The send methods add application messages to the send queue and the receive method retrieves application messages from the receive queue. In the future, any other reliable API can substitute the LSMP protocol API, if it has a class that implements a send and receive method. In addition, the LSMPPacket class instantiates the LSMPController class. It also instantiates the LSMPSender or LSMPReceiver class depending on whether the M2MP application is a sender or a receiver.

LSMPController – The LSMPController class sets the M2MP Channel and the M2MP Message Notifier. It implements the send and receive message method to send and receive messages using the M2MP API.

LSMPSender – This class implements the Reliable Send as described in Section 4.2.2.1. It retrieves the messages from the send queue, creates the ACKPending list and transmits the message using the LSMPController send method. It also listens for incoming ACK messages from the receivers using the LSMPController receive method.

LSMPReceiver – This class implements the Reliable Receive as described in Section 4.2.2.2. It listens for incoming messages from the senders using the LSMPController receive method. It detects and ignores missing or duplicate messages. For valid messages having the right sequence number, it sends an ACK using the LSMPController send method.

LSMPAsyncSender – This class has an asynchronous thread that is used to resend the current message being sent by the LSMP Sender, at periodic interval.

LSMPDatagramQueue – This class implements the data structure for the send and receive queue.

LSMPMessage – This is the LSMP Message object that is transmitted from the sender and receivers. It has set and get methods to specify and retrieve sender, receiver, sequence number and message originator information.

MessageInterface - This interface is extended by the LSMPMessage and the Application Message objects.

4.1.3.1 Send Communication Sequence

The high level sequence of events for sending a message by the Reliable layer is as follows

?? The Sender application uses the LSMPPacket.send() method to pass the message to the Reliable layer. The message is then inserted into the send queue.

?? The LSMPSender thread waits on the send queue. When a message is available in the send queue, the LSMPSender threads is notified. It retrieves and buffers the message and calls the LSMPController.sendMessage() method to send the message over the network. The LSMPSender thread then waits to receive an acknowledgement from all the know receivers.

?? The LSMPController thread is constantly listening from ACK messages and passes the messages to the LSMPSender class.

?? The LSMPAsyncSender periodically resends the message currently in the buffer.

The sequence diagram for the send communication is show below in Figure 7.

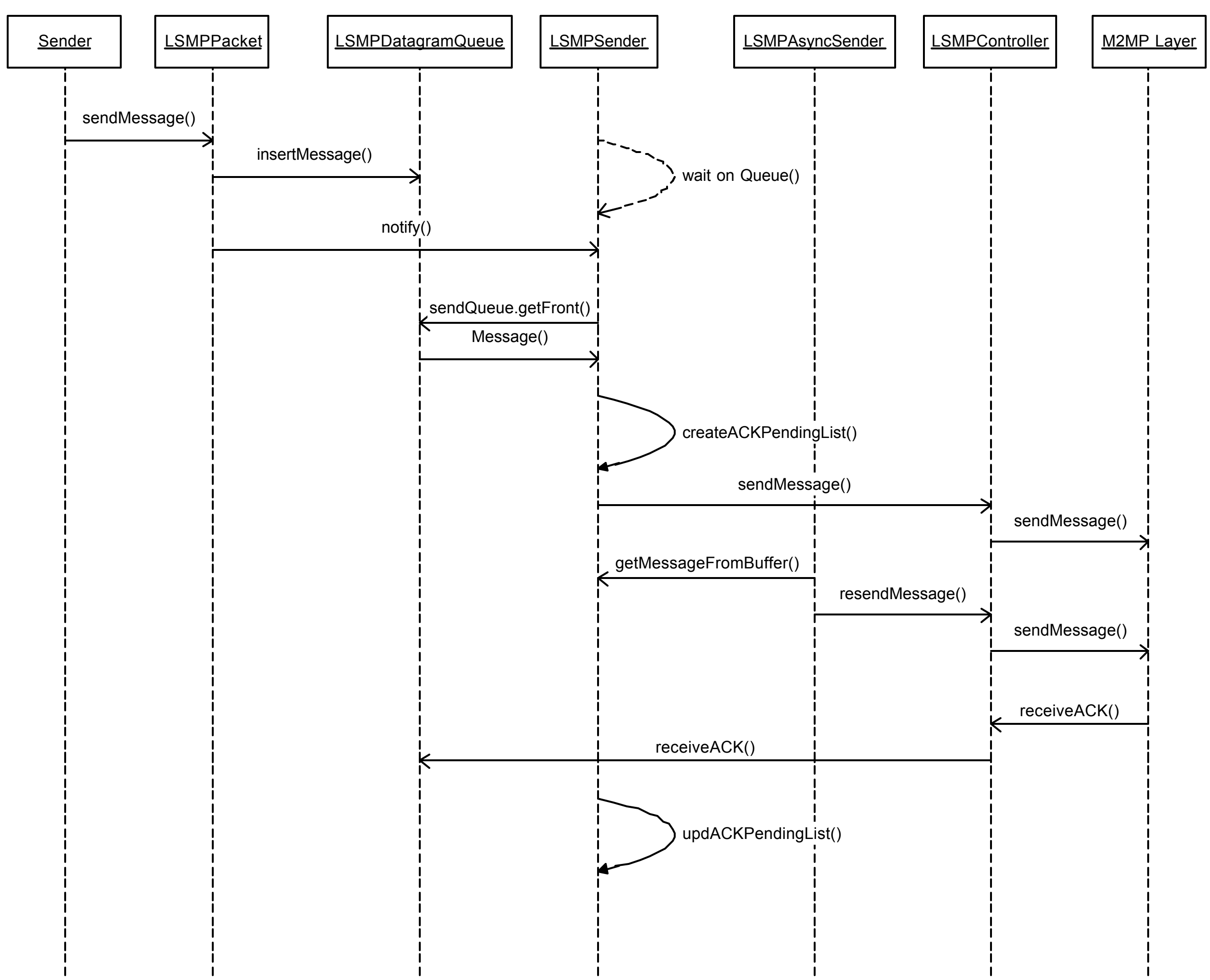


Figure 7: Reliable Send Communication Sequenc

4.1.3.2 Receive Communication Sequence

The high level sequence of events for receiving a message by the Reliable Layer is as follows.

- ?? The LSMPController class is constantly listening for messages and on receiving messages, passes them on to the LSMPReceiver class.
- ?? The LSMPReceiver class validates the message and inserts them into the receive queue. It also sends an acknowledgement of the message using the LSMPController.sendMessage method.
- ?? The LSMPPacket class waits on the receiver queue and on being notified of an available message, retrieves the message and passes it to the Receiver.

The sequence diagram for the receive communication is show below in Figure 8.

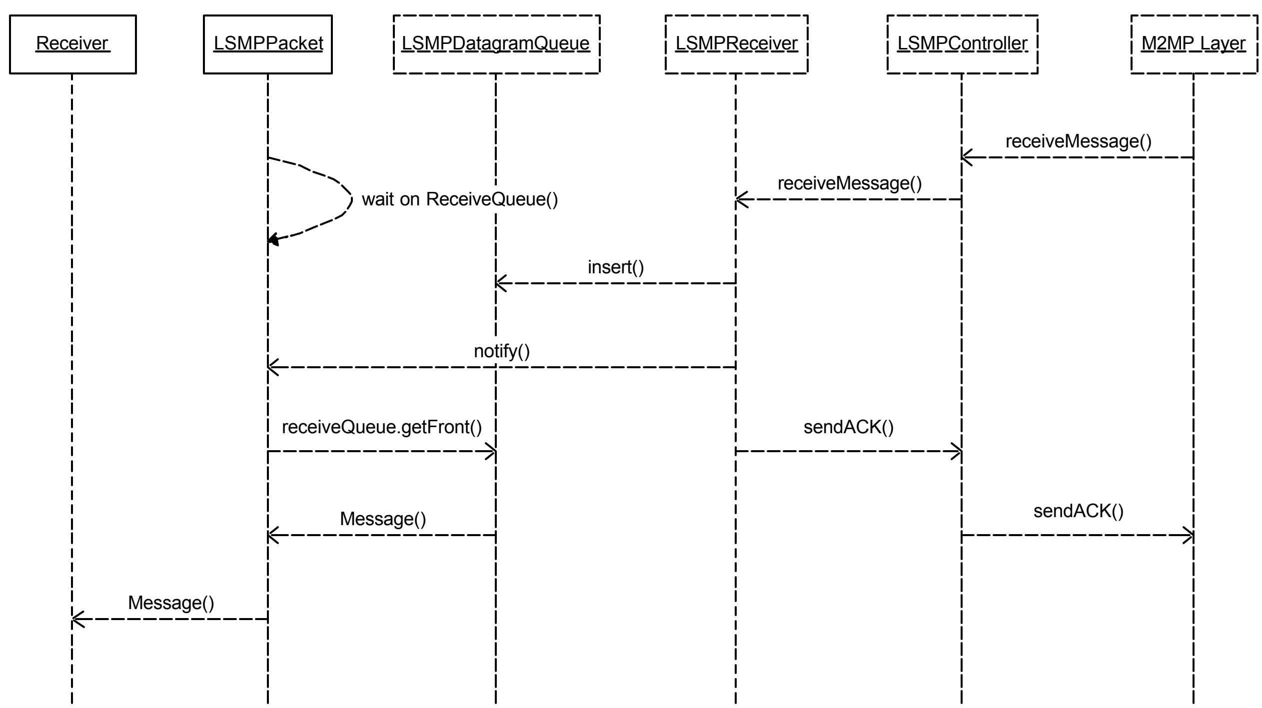


Figure 8: Reliable Receive Communication Sequence

4.2 Drop Packet Layer

4.2.1 Approach

The Drop Packet layer provides an ability to simulate an unreliable network. Based on the drop rate, which can be specified at runtime using a GUI interface, packets can be prevented from being sent to the members in the multicast group. The Drop Packet layer is implemented as a channel on top of the M2MP channel as shown in Figure 9. An advantage of the Channel implantation approach is that an application can choose to implement or not implement the Drop Packet layer [10].

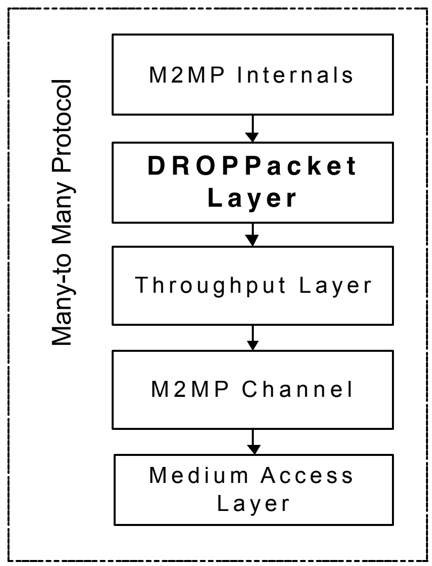


Figure 9: Drop Packet Layer Channel

4.2.2 Design Description

The design for the Drop Packet layer is as shown the below in Figure 10.

A packet is passed on from the Application layer to the Drop Packet layer. If the number of packets to be dropped is set to a non-zero number, the packet is not multicast to the members of the multicast group. This simulates a packet dropped on an unreliable network. The number of packets is set at runtime using a GUI interface and is rate based i.e. it specifies a given number of packets to be dropped over an interval of time. The design approach is independent of whether an application is a sender or receiver.

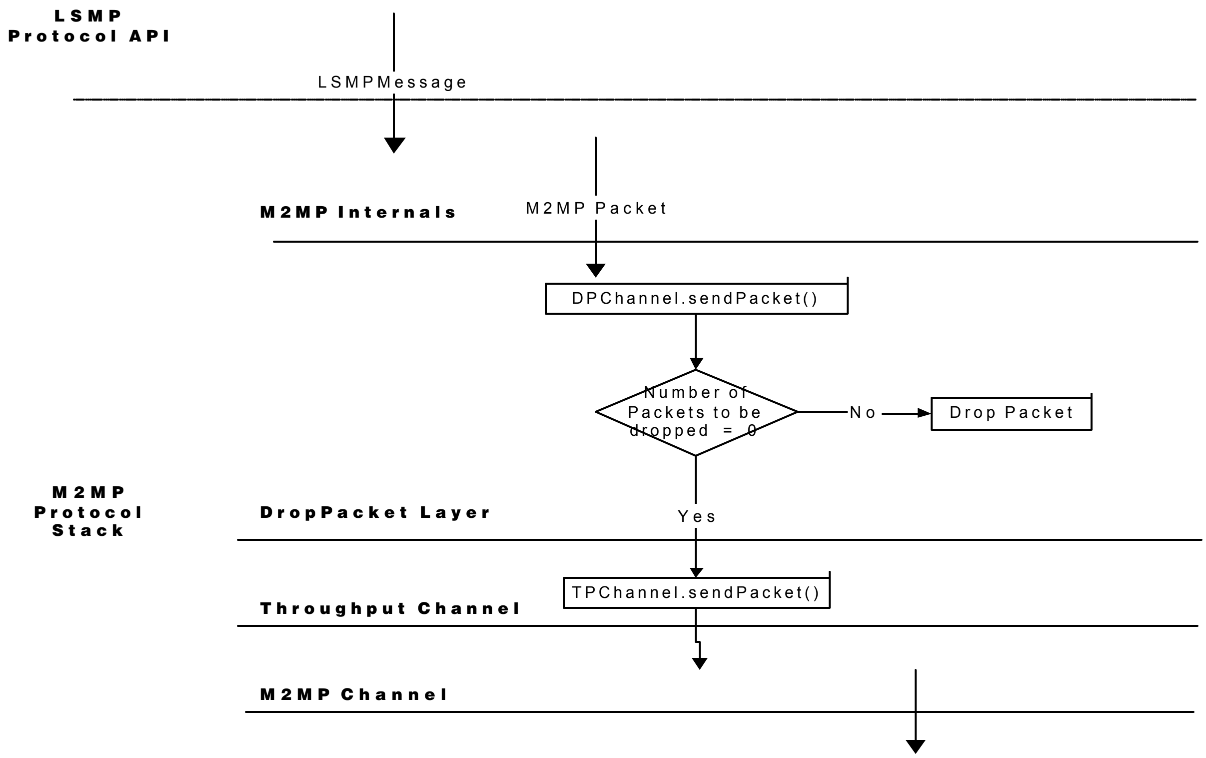


Figure 10: Drop Packet Layer Design

4.2.3 Communication Sequence

The following is a brief description of all the classes that make up the Drop Packet layer.

DPChannel – This class implements the Channel interface. It implements the transmitPacket and receivePacket methods. It transmits the packet if the value of number of packets to be dropped is set to zero.

DPManger – This class implements the Observable interface. It processes the packet received as a result of a change in the value of the number of packets to be dropped, which is set using the GUI interface. It also periodically sends the number of packets that have been dropped over a given interval of time, to the Remote Control application.

DPController - This class keeps track of the number of packets that need to be dropped. As the number set through the GUI interface is a drop packet rate, a timer keeps tracks of the time interval and resets the number of packets to be dropped to the initial value. It also keeps a count of packets that have been dropped over a given interval of time

DPException – This is an Exception class.

The sequence diagram for the Drop Packet layer is as shown below in Figure 11.

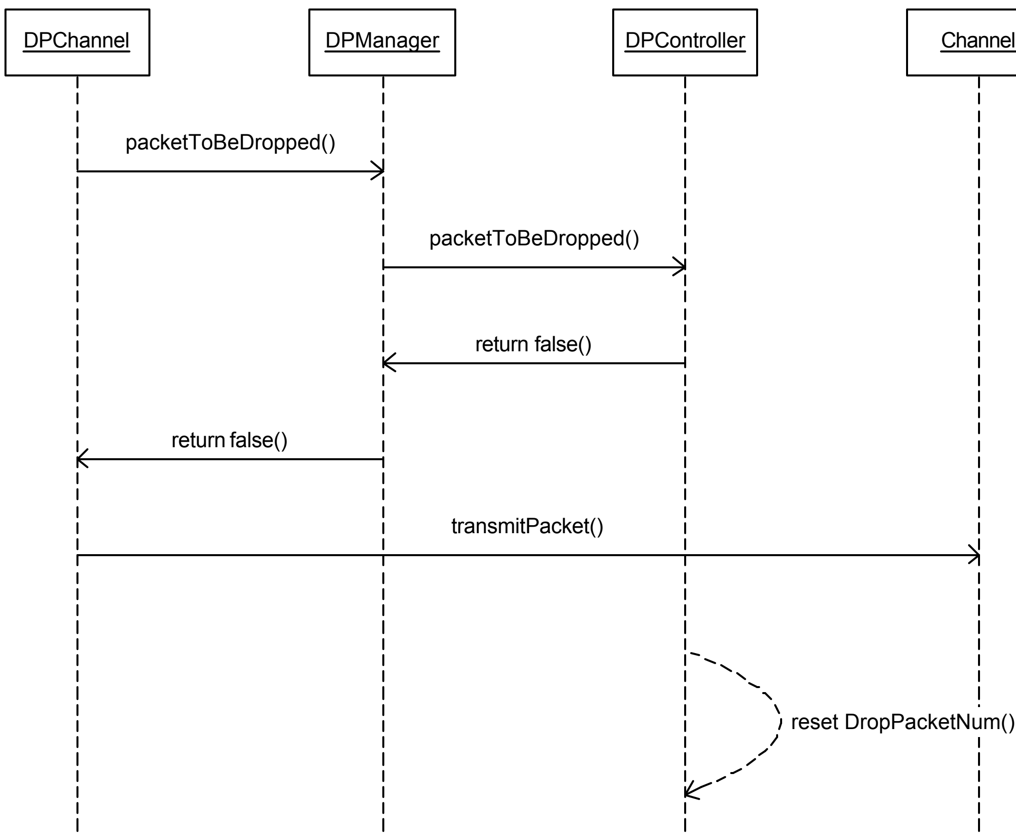


Figure 11: Drop Packet Layer Communication Sequence

The DPChannel class polls to see if the value of the number of packets to be dropped is zero or non-zero. If the value is zero, the DPChannel transmits the packet to the next Channel in the pipeline.

4.3 Throughput Layer

The Throughput layer was designed and implemented as part of an earlier work [10]. Please refer to [10] for a detailed description of the design and implementation of the Throughput Layer.

As part of this work the Throughput layer was integrated with the Drop Packet layer.

4.4 Remote Control Application

The Remote Control Application was designed and implemented as part of an earlier work [10]. Please refer to [10] for a detailed description of the design and implementation of the Remote Control Application. This section discusses the modification made to the Remote Control Application to support communication with the Drop Packet layer.

The Remote Control is a server process running on a known host and a port. For each client connecting to the server, the server creates a RemoteService thread [10] as shown below in Figure 12. The RemoteService Thread provides a GUI interface to set the send and receive delay and the drop packet rate and communicate the delay and drop rate information to the Throughput and Drop Packet layer.

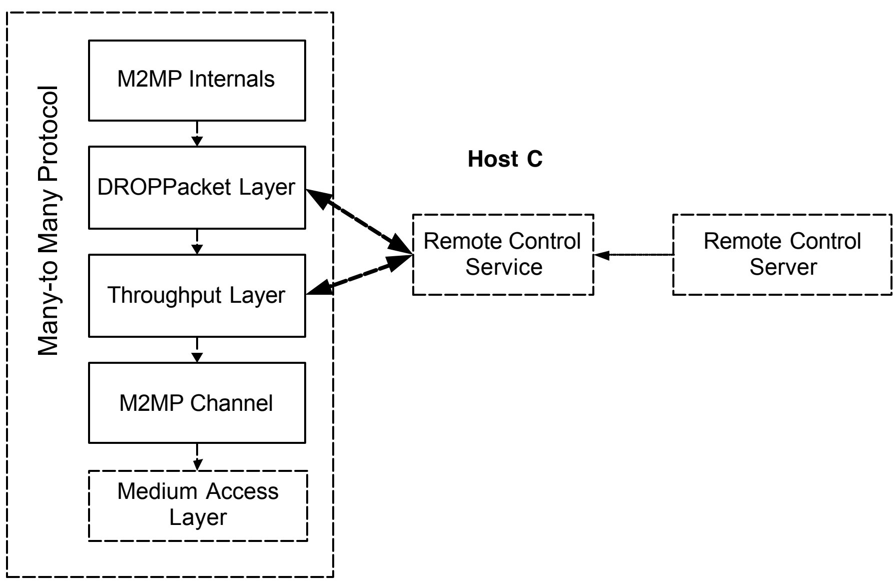


Figure 12: Remote Control Application

4.4.1 Design Description

The communication between the Remote Control Service and the M2M protocol layer is two ways – sending new delay and drop packet rate information to M2MP layer and receiving control information from the M2MP layer.

4.4.1.1 Remote Control Service Send

The design for the enhanced Remote Control application for sending packets to the M2MP layer is as shown below in Figure 13.

The send and receive delay and the drop packet rate is set using a GUI interface provided by the Remote Control Service. The information is encapsulated into a Message and sent across the network to the Communicator object. The Communicator object transmits and receives serialized messages from the Remote Control Service. In order for the Drop Packet layer and Throughput layer to be notified of new messages, the Observer design pattern was implemented. The DPManager and TPManager subscribe to the Communicator object and are notified when the Communicator receives new message. The message is processed by the DPManager class in the Drop Packet layer and the TPManager class in the Throughput layer, to retrieve the new delay and drop rate information.

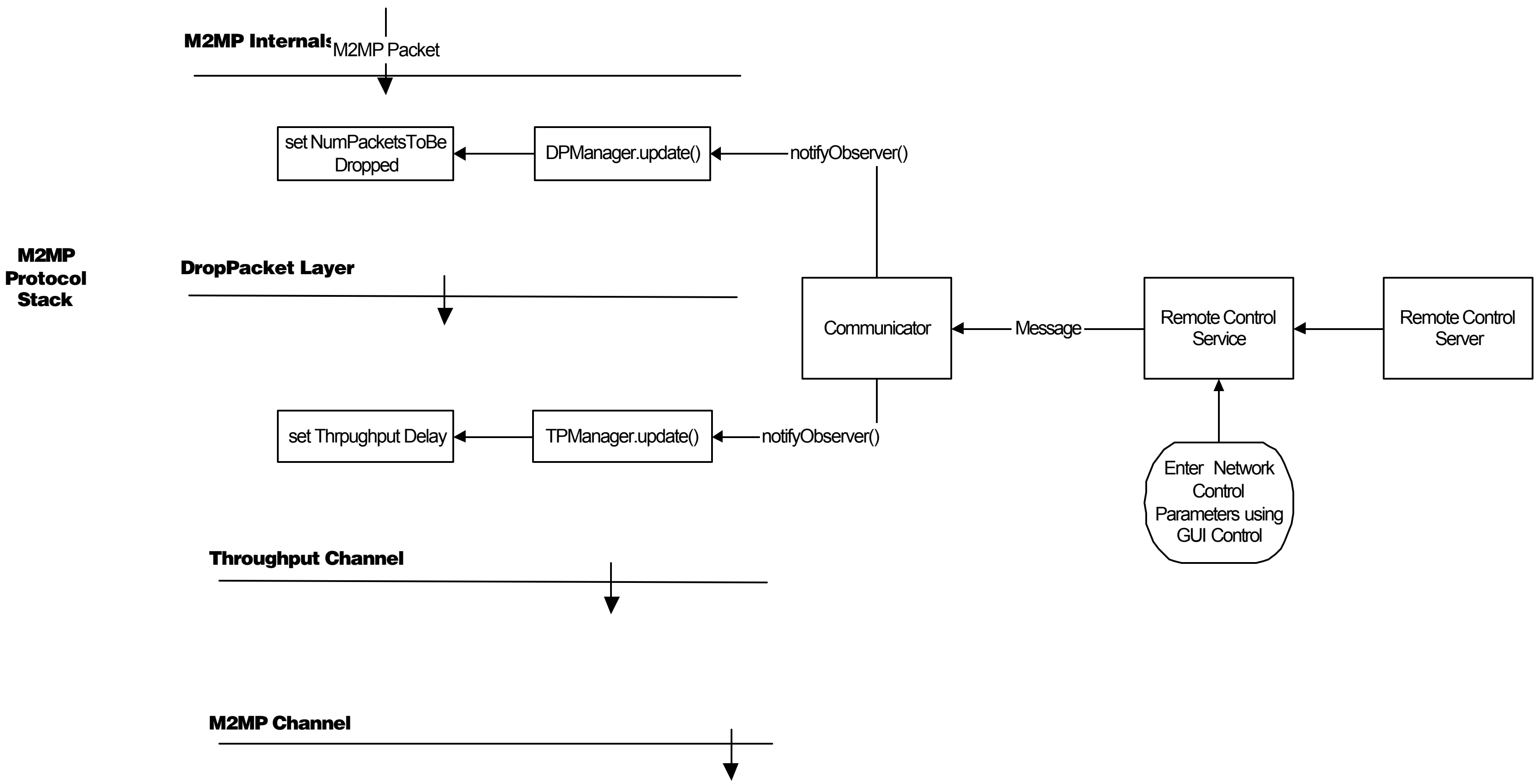


Figure 13: Remote Control Send Design

4.4.1.2 Remote Control Service Receive

The design for the enhanced Remote Control application for receiving packets from the M2MP layer is as shown below in Figure 14.

The Remote Control Service periodically receives information from M2MP layer. The Drop Packet layer keeps track of the number of packets dropped over a given period of time i.e. 10 seconds and sends a message to the Remote Control Service every 10 seconds. The Throughput Layer keeps track of the total number of packets sent and received over a given period of time i.e. 10 seconds and sends the message to the Remote Control Service every 10 seconds. The Remote Control Service processes the message received and plots the information on the GUI interface. The plot is updated every 10 seconds. It keeps track of the number of packets sent, received and dropped.

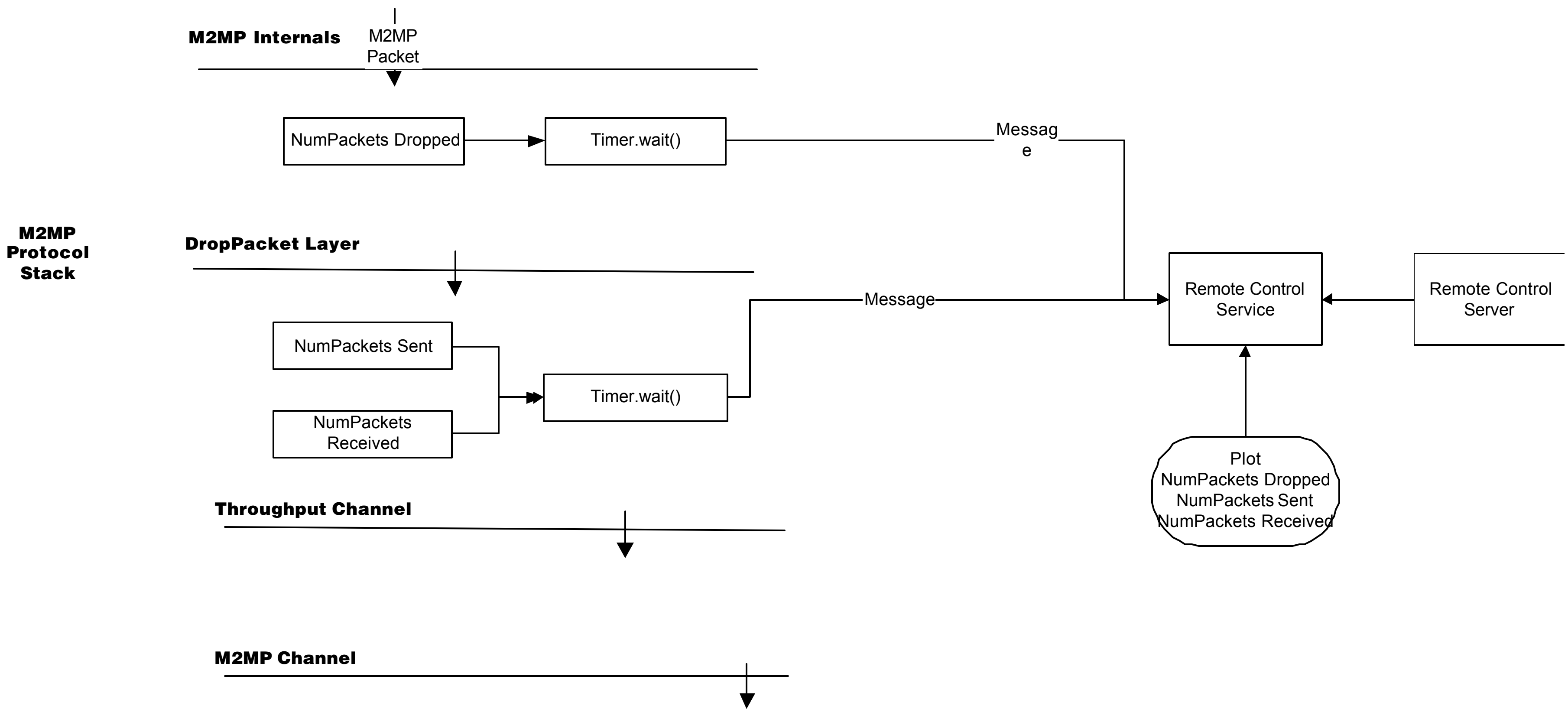


Figure 14: Remote Control Receive Design

4.4.1.3 GUI Interface

The GUI interface is as shown in below in Figure 15. The GUI Interface was modified to enable setting the drop packet rate and to plot receive and drop packet information, in addition to the send information.

The send and receiver delay and drop packet rate are set in the text boxes at the bottom of the GUI interface. As shown in the example in Figure 15, the receive delay is set to 100 milliseconds and the drop rate is set to 3 packets every 10 seconds. The GUI interface also plots the number of packets sent (blue), received (red) and dropped (green) at the sender. The plot is updated every 10 seconds. From the plot in example in Figure 15, the sender is sending about 16 packets every 10 seconds, receiving 20 packets every 10 seconds and about 3 packets every 10 seconds are being dropped on the network.

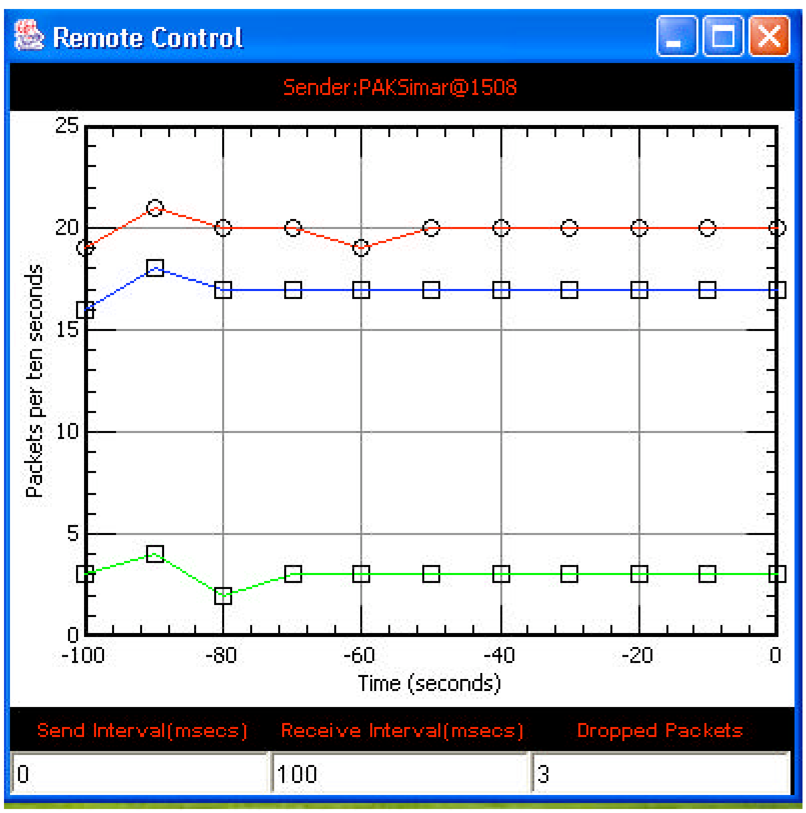


Figure 15: GUI Interface

4.4.2 Communication Sequence

4.4.2.1 Send Communication Sequence

The communication between the Remote Control Service and the M2MP layer is as shown below in Figure 16.

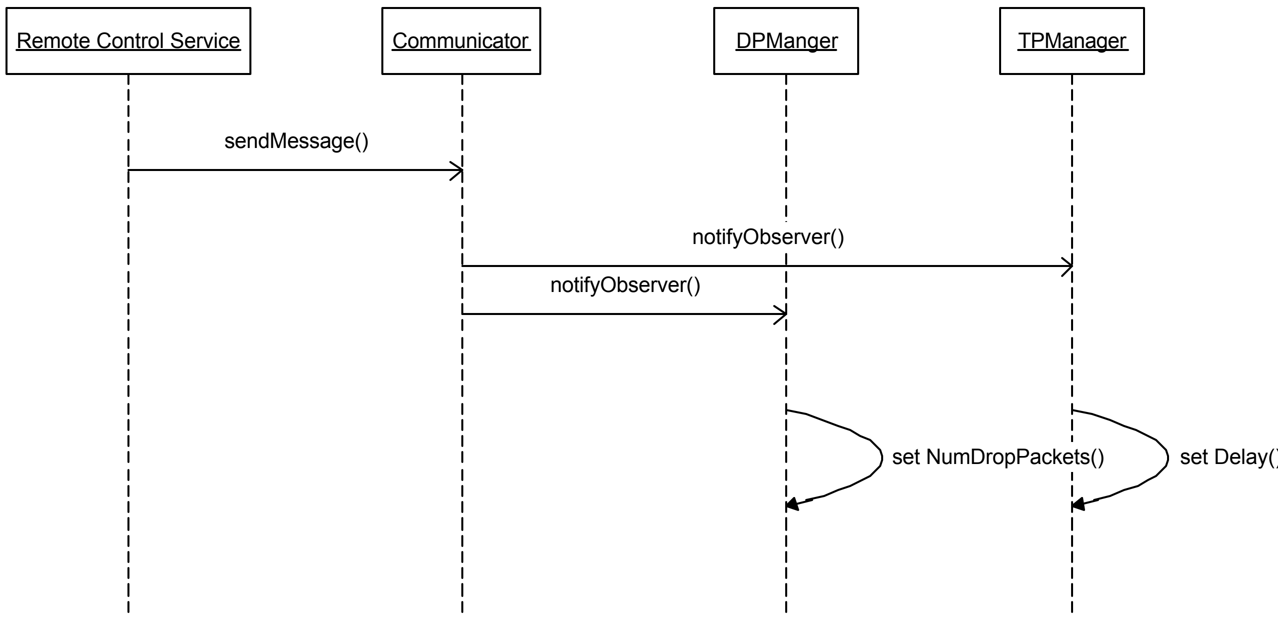


Figure 16: Remote Control Send Sequence

4.4.2.2 Receive Communication Sequence

The Remote Control Service Receive communication sequence is as shown below in Figure 17.

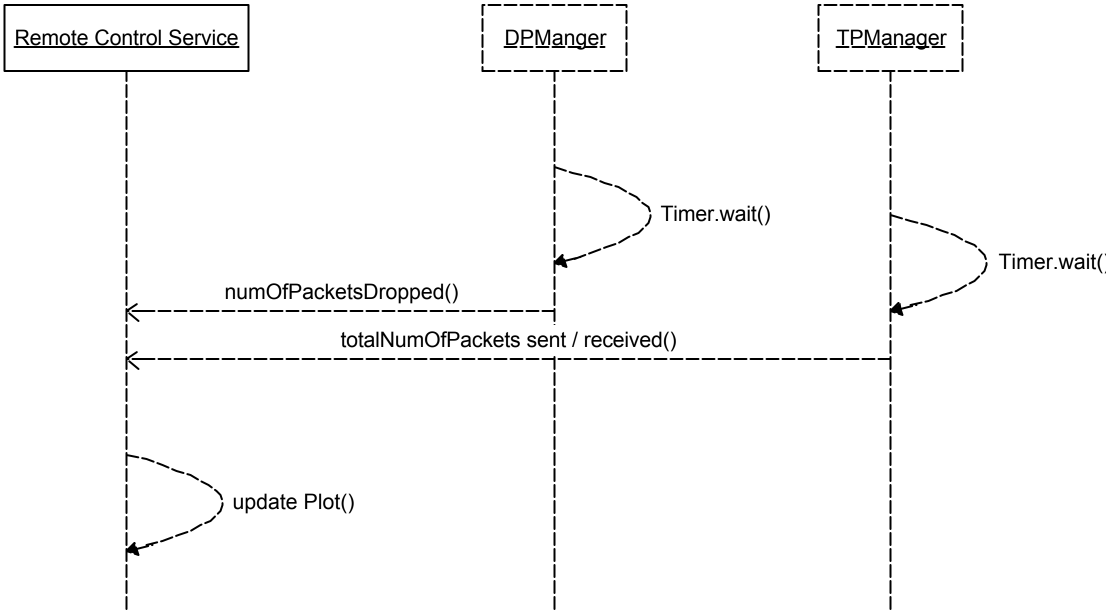


Figure 17: Remote Control Receive Sequence

5.0 Results

The aim of this study was to provide a framework for measuring and analyzing the performance of a reliable multicast service built on top of the M2M protocol stack. It was required to be able to conduct the analysis in reliable and unreliable network conditions.

In order to conduct the analysis, an existing M2MP application [10] was modified to use the LSMP service to send and receive packets. The Remote Service GUI interface was used to set the drop packet rate and the send/receive delay. The Remote Service then communicated the information to the Drop Packet and Throughput layer service to simulate unreliable network conditions.

Test data was collected by having each of the layers - Application, LSMP, Drop Packet and the Throughput layer – keep track of the number of packets processed in the respective layer and write the information to a log file. The information was written to the log file every 10 seconds.

Tests were conducted to measure the following for the LSMP service in reliable and simulated unreliable network conditions for one sender-to-many receivers and many senders-to-many receivers communication.

- ?? Throughput
- ?? Number of redundant and resent packets.

The tests were run in the RIT CS Computer Lab. The three machines used for the tests were holly.cs.rit.edu, hilly.cs.rit.edu and queer.cs.rit.edu. The setup for the test runs is detailed in the following section.

5.0 Reliable Network Conditions

The following tests were conducted under reliable network conditions.

5.0.1 Throughput

The maximum throughput attained by the LSMP sender was measured for one sender-to-one receiver, one sender-to-two receivers and two senders-to-two receivers scenarios. In addition, the number of application messages sent from the application layer to the LSMP layer, were also measured. In this study, one application message sent from the application layer to the LSMP layer corresponds to one LSMP packet sent from the LSMP layer to the M2MP layer.

The results of the test run for one sender-to-one receiver are as shown in Figure 18. The remote control server was running on hilly.cs.rit.edu. The sender was running on holly.cs.rit.edu and the receiver was running on queer.cs.rit.edu. The output of the test run is tabulated in Appendix A.

The results of the test run for one sender-to-two receivers are as shown in Figure 19. The remote control server was running on hilly.cs.rit.edu. The sender was running on hilly.cs.rit.edu and the receivers were running on queer.cs.rit.edu and holly.cs.rit.edu. The output of the test run is tabulated in Appendix B.

The results of the test run for two senders-to-two receivers are as shown in Figure 20. The remote control server was running on hilly.cs.rit.edu. The senders were running on hilly.cs.rit.edu and queer.cs.rit.edu. The receivers were running on queer.cs.rit.edu and holly.cs.rit.edu. The output of the test run is tabulated in Appendix C.

From the results in Figure 18, 19 and 20, the throughput of the Application layer for sending messages to the LSMP layer was consistent at 100 messages / second with minor dips for the one sender-to-one receiver and the one sender-to-two receivers scenario. For the two senders-to-two receivers scenario, the throughput still averaged 100 messages / second though the dips were little more pronounced, dropping to 95 messages / second.

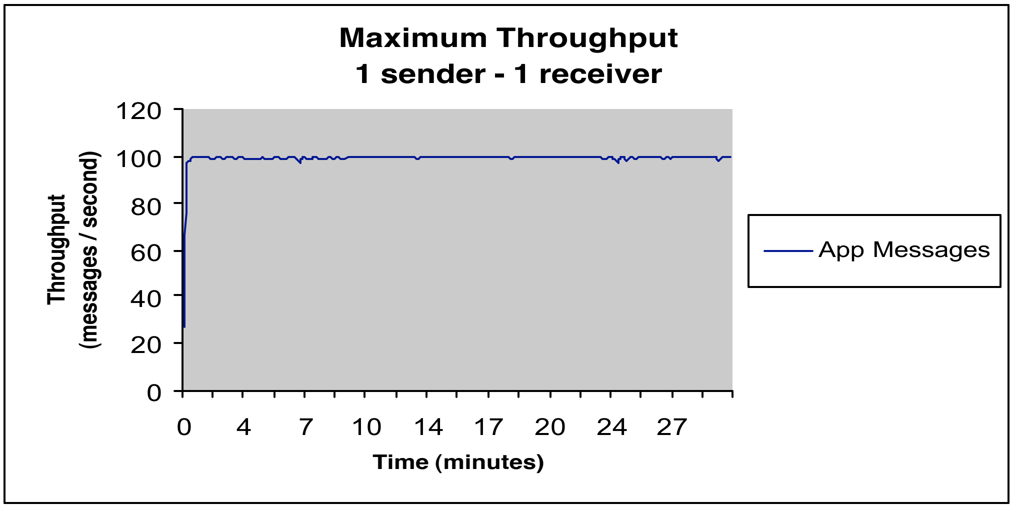
The throughput of the LSMP layer did not change when the number of receivers increased from one to two. In the one sender-to-one receiver and the one sender-to-two receiver scenarios the throughput averaged about 100 packets / second. But when the number of senders increased from one to two, the throughput decreased by about 30 percent to around 70 packets / second.

From above, the LSMP layer was able to keep up with the number of the Application messages being sent by the Application in the one sender scenario but not in the case of two senders.

In addition, the throughput results of the LSMP layer showed significant fluctuations in the number of packets sent - a drop in the number of packets sent, followed immediately by a peak. This behavior was observed in all three test scenarios, though in the case of two senders-to-two receivers the frequency of the fluctuations was much higher than in the case of one sender scenarios. The fluctuations can be attributed to the fact that the Application and the LSMP layer are writing to and retrieving messages from the same send queue structure. As access to the send queue is synchronized, the LSMP layer would have to wait while the Application layer was writing messages to the send queue. This would result in a drop in the number of LSMP packets sent. After the Application layer is done writing to the send queue, the LSMP layer would immediately be able to process all the packets in the send queue. This would result in immediate increase or peak in the number of packets sent by the LSMP layer. The behavior can also be observed by reviewing the output of the test run in Appendix A, B and C, specifically the value of the column *LSMP-App*.

The reason for the higher frequency of the fluctuations in the case of two senders-to-two receivers scenario could be due to the size of the send queue. In this scenario, the LSMP layer was unable to keep up with the number of Application messages being sent by the Application. As a result the send queue increased in size over time. With the increase in the size of the send queue, the time to write to and retrieve from the send queue would increase incrementally resulting in higher frequency of fluctuations.

In Figure 19, for the one sender-to-two receivers scenario, initially the throughput was zero, and then peaked to high value of around 150 packets / second. This was because there was a small delay in starting the receiver applications. As a result, the sender application was not able to receive acknowledgement for packets sent and the throughput was zero. When the receiver applications came online, the sender was immediately able to process all the saved packets in the sender queue, resulting in the high throughput. This was expected behavior. The behavior can also be observed by reviewing the output of the test run in Appendix B.



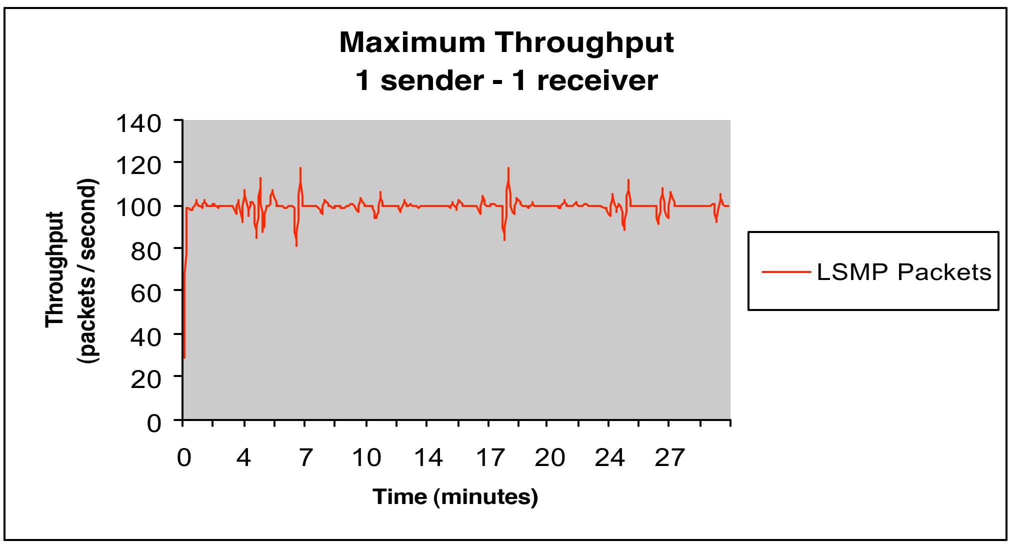
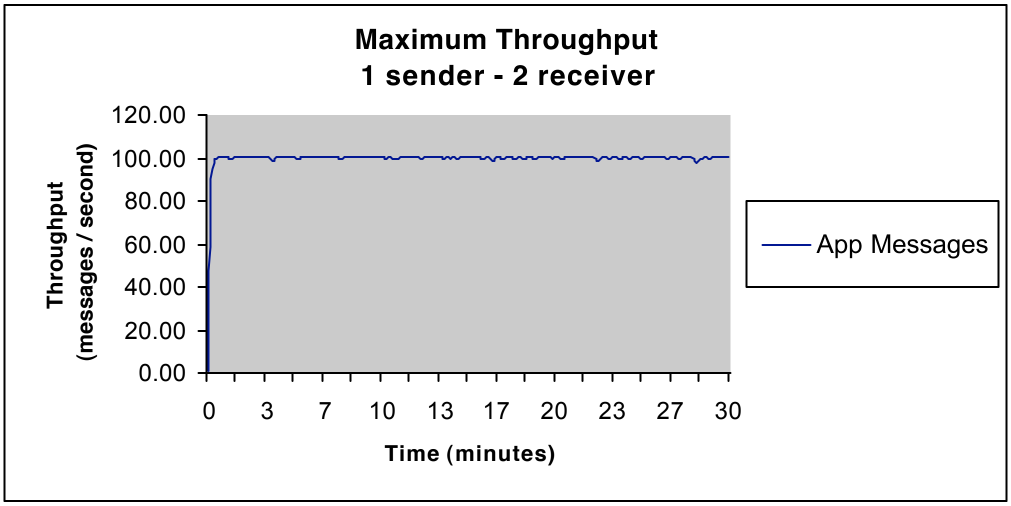


Figure 18: Max Throughput for App Messages and LSMP Packets for 1 sender - 1 receiver in reliable network



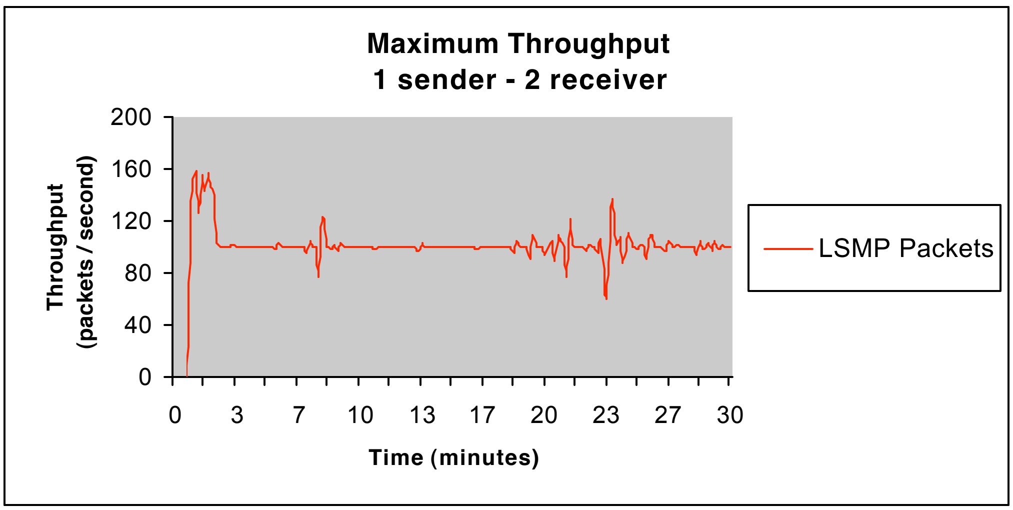


Figure 19: Max Throughput for App Messages and LSMP Packets for 1 sender - 2 receivers in reliable network

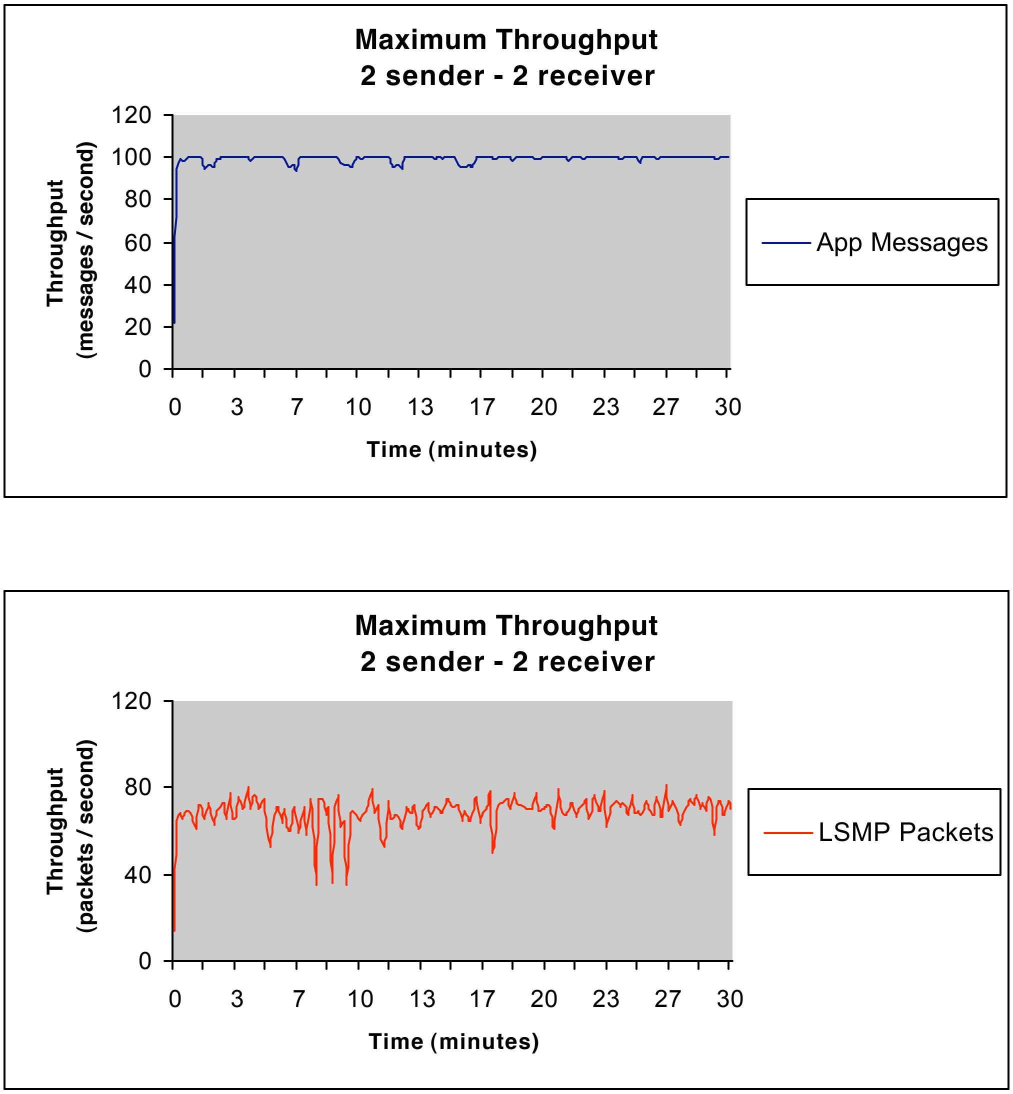


Figure 20: Max Throughput for App Messages and LSMP Packets for 2 senders - 2 receivers in reliable network

5.0.2 Resent Messages

The sender periodically resends messages to the receivers if the receivers fail to acknowledge the message in a given interval of time. A high number of resent messages might indicate an unreliable network link. The results for the number of resent messages are as shown in Figure 21. The number of resent messages were the same for all the scenarios at 50 packets / second. The resend interval for the tests in reliable conditions was 1 packet / 10 milliseconds.

Based on the design of the application the default behavior of the LSMP layer is to resend about 50 packets / second. As all the scenarios were run in reliable network conditions, it is to be expected that the number of resent packet rate is 50 packets / second in all cases.

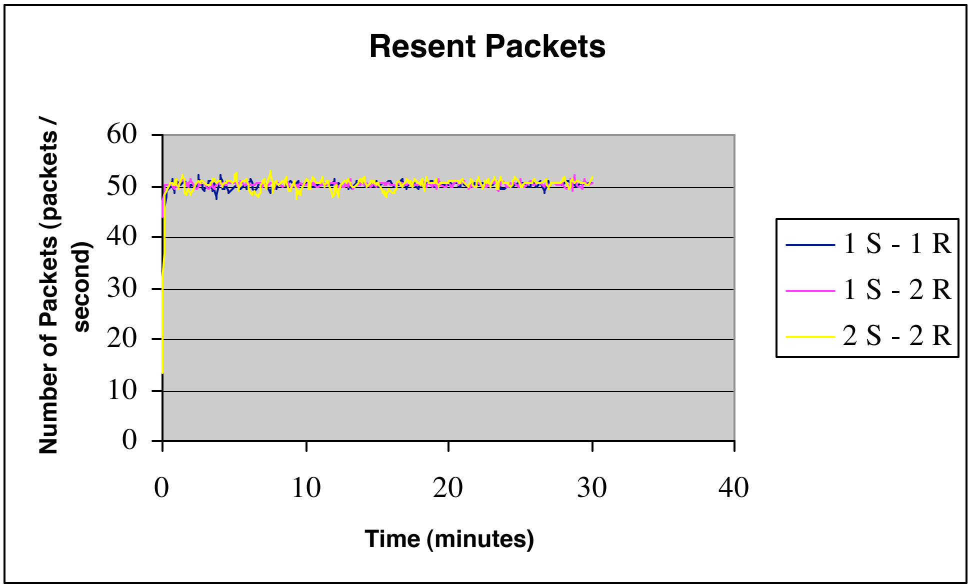


Figure 21: Resent messages in a reliable network

5.1 Simulated Unreliable Network Conditions

The maximum throughput attained by the LSMP sender was measured for one sender-to-one receiver, one sender-to-two receivers and two senders-to-two receivers scenarios. In addition, the number of application messages sent from the application layer to the LSMP layer, were also measured. In this study, one application message sent from the application layer to the LSMP layer corresponds to one LSMP packet sent from the LSMP layer to the M2MP layer

In order to simulate unreliable network conditions, the drop rate was specified using the GUI interface.

5.1.1 Throughput

The results of the test run for one sender -to-one receiver are as shown in Figure 22. The remote control server was running on hilly.cs.rit.edu. The sender was running on holly.cs.rit.edu and the receiver was running on queer.cs.rit.edu. The output of the test run is tabulated in Appendix D.

The results of the test run for one sender-to-two receivers are as shown in Figure 23. The remote control server was running on hilly.cs.rit.edu. The sender was running on hilly.cs.rit.edu and the receivers were running on queer.cs.rit.edu and holly.cs.rit.edu. The output of the test run is tabulated in Appendix E.

The results of the test run for two senders-to-two receivers are as shown in Figure 24. The remote control server was running on hilly.cs.rit.edu. The senders were running on hilly.cs.rit.edu and queer.cs.rit.edu. The receivers were running on queer.cs.rit.edu and holly.cs.rit.edu. The output of the test run is tabulated in Appendix F.

The drop rate for all the test runs was set to 10 packets / second. Also, the interval of time for resending the packets asynchronously was set to 10 milliseconds.

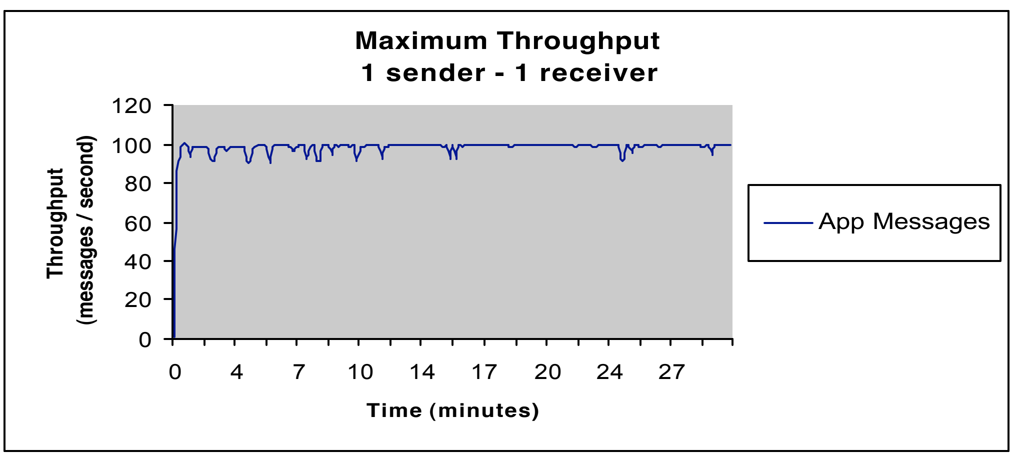
From the results in Figure 22, 23 and 24, the throughput of the Application layer was similar to the throughput observed in reliable network conditions. This was expected behavior as dropping of packets on the network would not have any impact on the ability of the Application layer to send messages to the LSMP layer.

The results of the throughput of the LSMP layer in the one sender-to-one receiver scenario as shown in Figure 22 indicated that for the most part, the LSMP layer was able to keep up with the number of application messages being sent by the Application layer. But due to dropping of packets on the network, the LSMP layer had to resend messages resulting in the increase in the size of the send queue. As a result, a higher frequency of fluctuations was noticed corresponding to increase and then a drop in the size of the send queue.

In the case of the one sender-to-two receivers scenario as shown in Figure 23, the LSMP layer was not able to keep up with the number of application messages being sent by the Application layer, due to dropping of packets on the network. As a result the size of the send queue increased with time resulting in the higher frequency of fluctuations when compared to the one sender-to-two receiver scenarios in the reliable network conditions.

In case of the two senders-to-two receivers scenario as shown in Figure 24, the LSMP layer could not keep up with the number of messages being sent by the Application. The throughput of the LSMP layer dropped to about 55 packets / seconds i.e. a drop of about 45 percent. After about 14 minutes into the test run, the throughput increased to a little above 100 packets / second as shown in Appendix F. The reason for this was that the second sender application failed with an *Out-of-Memory* error. The caused the scenario to be similar to the one sender-to-two receivers scenario and hence the increase in the throughput.

The test run for the two sender-to-two receiver scenario was run twice and demonstrated the same behavior both the times.



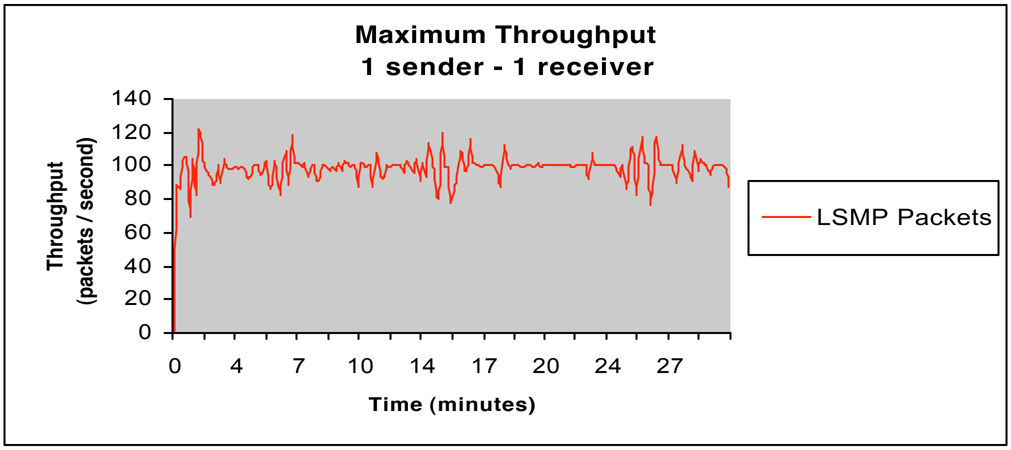
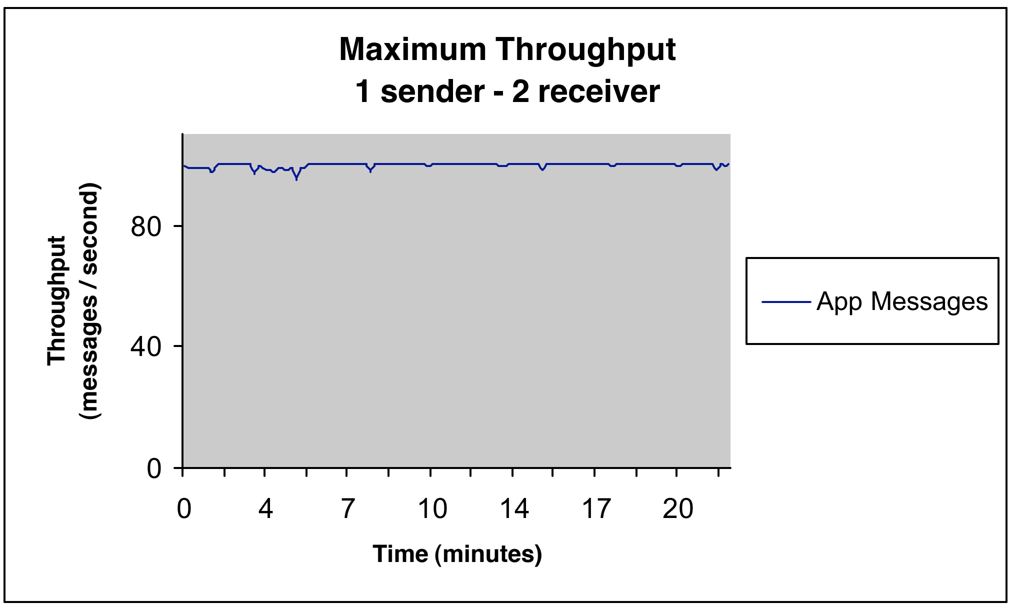


Figure 22: Max Throughput for App Messages and LSMP Packets for 1 sender - 1 receiver in unreliable network



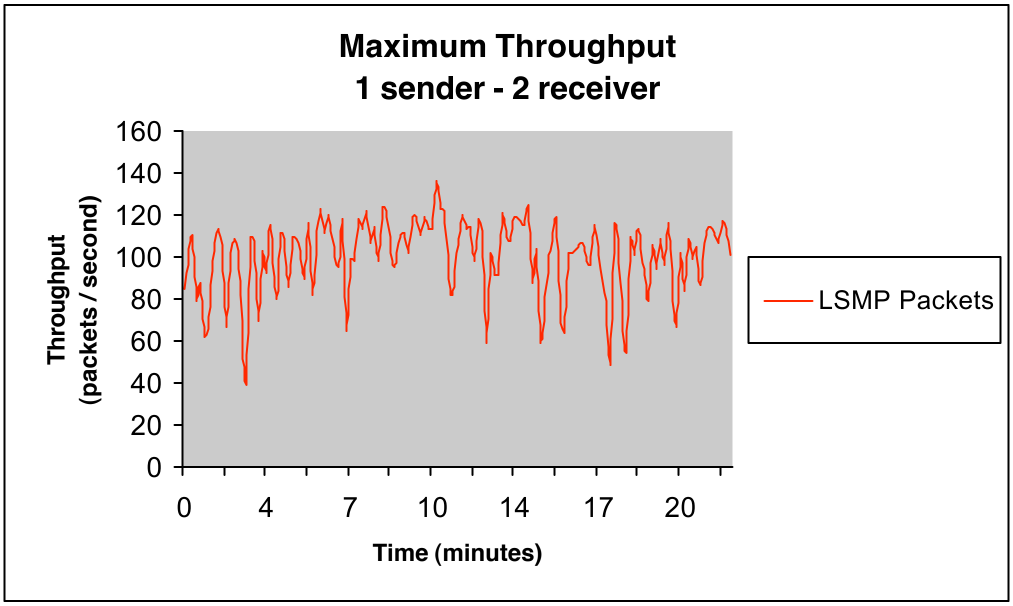
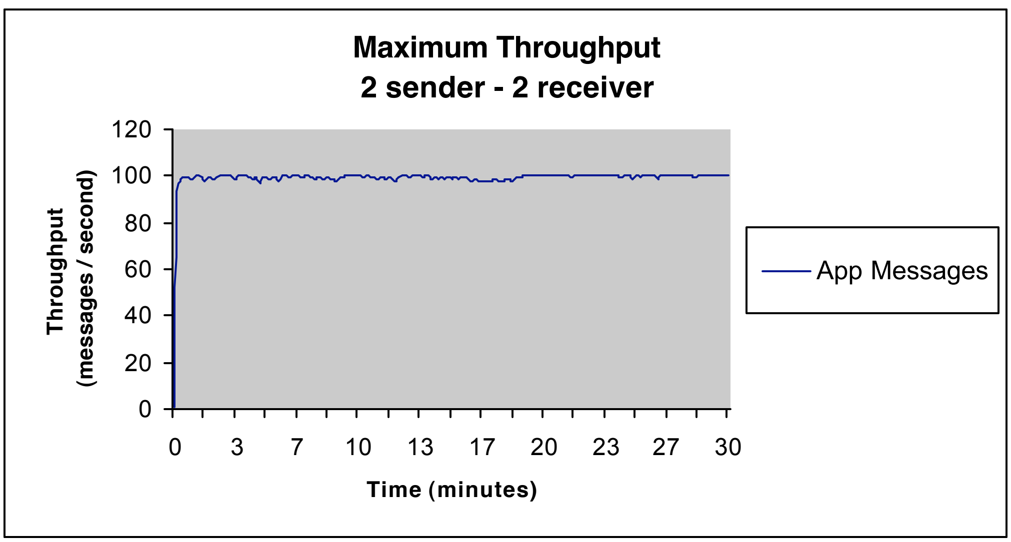


Figure 23: Max Throughput for App Messages and LSMP Packets for 1 sender - 2 receivers in unreliable network



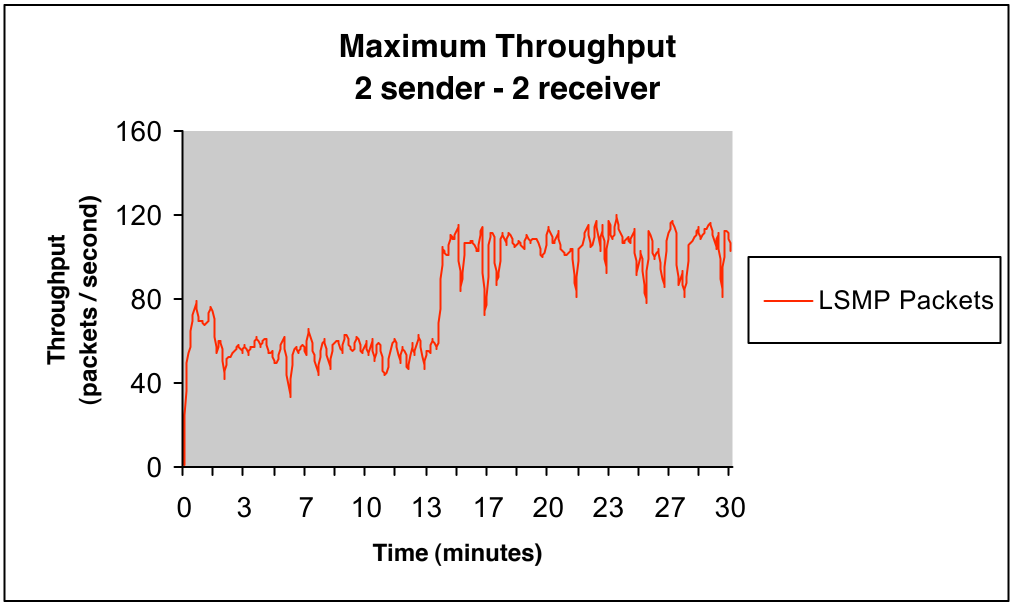


Figure 24: Max Throughput for App Messages and LSMP Packets for 2 sender - 2 receivers in unreliable network

5.1.2 Resent Messages

The results for the number of resent messages in unreliable network for all the scenarios are as shown in Figure 25. The number of resent messages were the same for all the scenarios at 50 packets / second. The resend interval for the tests in reliable conditions was 1 packet / 10 milliseconds. This would indicate that in even in unreliable network conditions where the drop rate was 10 packets / second, the number of packets resent by the LSMP layer was not greater than the default resent rate of 50 packets / seconds for the all the scenarios.

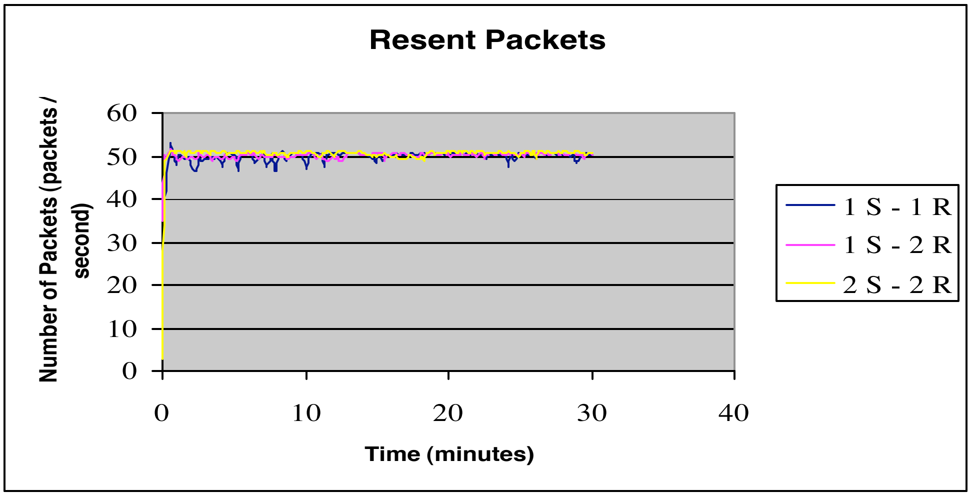


Figure 25: Resent message in unreliable network

6.0 Conclusions

6.0 LSMP In Reliable Network

The LSMP layer was able to keep up with the number of messages being sent by the Application layer in the case of one sender-to-one receiver and one sender-to-two receiver scenario. With the addition of a sender in the two senders-to-two receivers scenario, the maximum throughput attained by the LSMP layer decreased.

The above can be attributed to the fact that for additional receivers, the sender had to keep state information for each receiver and process acknowledgements from each receiver. In the case where there were multiple senders, each sender had also to process and discard the packets sent from other senders. The LSMP senders are single threaded. They process the packets received in a sequential order. With the increase in the number of senders and receivers, the workload for a given sender increases resulting in decreasing throughput. In addition, a high frequency of fluctuations in the throughput was observed with increase in senders and receivers.

This would indicate that for long running sessions and for sessions with a large number of participants, the LSMP layer would not scale well.

The number of packets resent by the LSMP layer was the default application resent rate. This was expected in reliable network conditions where, there was no send / receives delay or loss of packets.

6.1 LSMP In Unreliable Network

The maximum throughput attained in unreliable network showed a trend similar to the trend observed in the reliable network. The throughput decreased with the addition of senders and receivers. The application required higher computing resources for running in simulated unreliable network conditions as is evident from the *Out-of-Memory* error observed while running in the two sender-to-two receivers scenario.

On comparing the output of the test run in reliable and unreliable network conditions, the throughput attained in unreliable network conditions was lower than the throughput in reliable network conditions. Also, as result of lower throughput and the corresponding increase in send queue size, the frequency of throughput fluctuation was much higher in unreliable network conditions. This was expected behavior.

The number of packets resent by the LSMP layer in unreliable network conditions was the default application resent rate. Unreliable network conditions did not cause an increase in the number of resent packets when compared to reliable network conditions. This might not be the case if the number of dropped packets is increased from 10 packets / second to a higher drop rate.

7.0 Limitations

The LSMP sender requires the information on the number of receivers participating in a session in order to create the ACK pending list. This limits the ability of receivers to late-join in a session. Also, the number of receivers cannot be changed dynamically.

There was no upper limit set on the Send and Receive queues. This has a potential for memory contention in long running or large sessions. This was evident in the two senders-to-two receivers scenario in unreliable network conditions.

8.0 Future Work

As part of this study, a simple sender-based reliable protocol was implemented. For future work, a receiver-based reliable protocol can be designed and implemented.

The LSM protocol supported late-join partially. Senders could join anytime during the session. The protocol can be updated to allow late-join for receivers. LSMP does not cache data persistently. The protocol can be modified to allow persistent storage of data.

The LSM protocol can be updated to support fault tolerance. In its current implementation, if the receiver disengages or become unavailable, the senders would not transmit any new data packets to the receivers.

The LSMP layer supported sender to receiver communication along the lines of LBRM and TRAM. The protocol can be updated to support collaborative applications where all participants can be both senders and receivers.

9.0 References

[1] Chiu, Dah-Ming., Hurst, Stephen., Kadansky, Miriam.,Wesley, Joseph., "TRAM: A tree based Reliable Multicast Protocol", SML TR-98-66. 1998. Electronically available on the Internet at http://www.experimentalstuff.com/sunr/techrep/1998/abstract-66.html. Downloaded March 10th, 2004.

[2] Floyd, Sally, Jacobson, Van, Liu, Ching-Gung, McCanne, Steven., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEEE/ACM Transactions on Networking. Electronically available on the Internet at http://citeseer.ist.psu.edu/495705.html. Downloaded April 2003.

[3] Gemmell, Jim., Leibeherr, Jorg., Bassett, Dave., "In Search of an API for Scalable Reliable Multicast", Technical Report MSR-TR-97-17, Microsoft Corporation. Electronically available on the Internet at http://microsoft.com/pub/tr/tr9717.ps
Downloaded April 2003.

[4] Holbrook, Hugh., W., Singhal, Sandeep., K., Cheriton, David., R., "Log-Based Receiver-Reliable for Distributed Interactive Simulation," Department of Computer Science, Stanford University. 1995. Electronically available on the Internet at http://citeseer.ist.psu.edu/holbrook95logbased.html. Downloaded April 2003.

[5] Kaminsky, Alan., Bischof, Hans-Peter., "The Many-to-Many Protocol Library,". Electronically available on the Internet at http://www.cs.rit.edu/~anhinga/m2mp.shtml. Downloaded February 5th, 2003.

[6] Kaminsky, Alan., Bischof, Hans-Peter., "Many-to-Many Invocation: A new framework for building collaborative applications in ad hoc networks," CSCW 2002 Workshop on Ad Hoc Communication and Collaboration in Ubiquitous Computing Environments, New Orleans, Louisiana, USA, November 2002.

[7] Kaminsky, Alan., Bischof, Hans-Peter., " Many-to-Many Invocation: A new object oriented paradigm for ad hoc collaborative systems," 17th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA 2002), Onward! track, Seattle, Washington, USA, November 2002.

[8] Kurose, James., F., Ross, Keith., W., "Computer Networking, A Top-Down Approach Featuring the Internet," Addison Wesley, 2001.

[9] Levine, David., L., Schmidt, Douglas., C., "Introduction to Patterns and Frameworks," Department of Computer Science, Washington University, St. Louis. 2003. Electronically available on the Internet at http://www.cs.wustl.edu/~schmidt/PDF/patterns-intro4.pdf. Downloaded March 20th, 2003.

[10] Narrepa, Ravi., "Throughput Control for a Many-to-Many System," Masters Thesis, Rochester Institute of Technology, 2002.

[11] Prakash, Ravi., Schiper, Andre., Mohsin, Mansoor., " Reliable Multicast in Mobile Networks," The University of Texas at Dallas, Richardson, Texas. Electronically available on the Internet at http://www.utdallas.edu/~mmohsin/publications/ReliableMulticast.pdf. Downloaded April 20th, 2003.

[12] Rosenzweig, Phil., Kadansky, Miriam., Hanna, Steve., "The Java Reliable Multicast Service: A Reliable Multicast Library", TR-98-68. 1998. Electronically available on the Internet at http://www.experimentalstuff.com/sunr/techrep/1998/abstract-68.html. Downloaded March 10th, 2004.

[13] Liao, Tie., " Light-weight Reliable Multicast Protocol.," INRIA. France. Electronically available on the Internet at http://webcanal.inria.fr/lrmp/index.html. Downloaded March 18th, 2004.

Appendix A - Output for Reliable One Sender-to-One Receiver Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets
0.17	26	29	2	71	42
0.33	97	98	1	147	49
0.50	98	97	0	147	50
0.67	100	98	-2	149	51
0.83	100	102	2	151	49
1.00	100	100	0	152	51
1.17	100	98	-2	148	50
1.33	100	102	2	153	51
1.50	100	99	-1	150	51
1.67	99	100	0	150	51
1.83	100	101	2	151	50
2.00	100	99	-2	148	50
2.17	100	100	0	151	50
2.33	98	99	1	149	50
2.50	100	100	-1	152	52
2.67	100	100	0	151	51
2.83	100	100	0	150	50
3.00	99	96	-4	145	49
3.17	100	103	3	154	51
3.33	100	93	-7	144	51
3.50	99	106	7	156	49
3.67	100	95	-4	146	51
3.83	99	102	3	149	48
4.00	99	100	1	152	52
4.17	99	85	-14	135	50
4.33	98	113	14	162	50
4.50	100	88	-12	139	51
4.67	98	99	1	148	49
4.83	98	101	3	150	50
5.00	98	106	8	156	50
5.17	100	100	0	150	50
5.33	100	100	0	151	51
5.50	100	100	0	149	50
5.67	99	99	0	149	50
5.83	100	99	-1	149	50
6.00	100	99	-1	149	49
6.17	100	99	-1	150	51
6.33	99	81	-18	131	51
6.50	97	118	21	166	49
6.67	100	100	0	151	50
6.83	99	99	0	149	50
7.00	99	99	0	149	50
7.17	100	99	0	150	50
7.33	100	100	0	150	50
7.50	99	99	0	147	49

7.67	99	96	-4	147	51
7.83	99	103	4	153	50
8.00	100	100	0	149	49
8.17	100	100	0	150	50
8.33	99	98	-1	148	50
8.50	100	100	1	151	50
8.67	100	100	0	150	50
8.83	99	99	0	148	50
9.00	100	99	0	150	51
9.17	100	100	0	150	50
9.33	100	101	1	151	50
9.50	100	100	0	151	51
9.67	100	97	-3	147	50
9.83	100	103	3	153	50
10.00	100	100	0	151	51
10.17	100	100	0	151	51
10.33	100	100	0	150	50
10.50	100	100	0	150	50
10.67	100	94	-6	144	50
10.83	100	106	6	156	50
11.00	100	100	0	150	50
11.17	100	100	0	151	50
11.33	100	100	0	150	50
11.50	100	100	0	150	50
11.67	100	100	0	150	50
11.83	100	100	0	150	50
12.00	100	97	-4	148	51
12.17	100	103	2	152	50
12.33	100	100	0	151	51
12.50	100	101	1	151	50
12.67	100	100	0	150	50
12.83	100	100	0	151	51
13.00	98	98	0	148	50
13.17	100	100	0	151	51
13.33	100	100	0	150	50
13.50	100	100	0	151	51
13.67	100	100	0	150	51
13.83	100	100	0	151	51
14.00	100	100	0	151	51
14.17	100	100	0	150	51
14.33	100	100	0	150	50
14.50	100	100	0	151	51
14.67	100	98	-2	149	51
14.83	100	102	2	152	50
15.00	100	98	-2	149	51
15.17	100	103	3	153	50
15.33	100	100	0	150	51
15.50	100	100	0	151	50
15.67	100	100	0	150	51
15.83	100	100	0	151	51

16.00	100	99	-1	150	51
16.17	100	100	0	150	50
16.33	100	96	-4	147	50
16.50	100	104	4	154	50
16.67	100	100	0	151	50
16.83	100	100	-1	151	51
17.00	100	101	1	150	50
17.17	100	100	0	151	51
17.33	100	100	0	150	51
17.50	100	99	-1	149	50
17.67	100	83	-17	134	51
17.83	100	117	17	166	49
18.00	99	100	1	151	51
18.17	100	96	-4	146	50
18.33	100	104	4	154	50
18.50	100	101	0	151	51
18.67	100	100	0	151	50
18.83	100	100	0	151	51
19.00	100	99	-1	149	50
19.17	100	102	2	152	50
19.33	100	100	0	151	51
19.50	100	100	0	150	50
19.67	100	100	0	150	50
19.83	100	100	0	150	50
20.00	100	100	0	150	50
20.17	100	100	0	151	51
20.33	100	100	0	150	50
20.50	100	100	0	151	51
20.67	100	99	-1	150	50
20.83	100	99	-2	149	50
21.00	100	102	2	153	51
21.17	100	100	0	150	50
21.33	100	100	-1	149	50
21.50	100	99	-1	149	50
21.67	100	102	1	152	51
21.83	100	100	0	150	50
22.00	100	100	-1	150	51
22.17	100	101	1	151	50
22.33	100	100	0	151	51
22.50	100	100	0	150	50
22.67	100	100	0	150	50
22.83	100	100	0	150	50
23.00	99	99	0	149	50
23.17	99	99	0	149	51
23.33	100	99	-2	149	51
23.50	100	95	-5	145	50
23.67	99	105	7	155	50
23.83	97	97	0	146	49
24.00	100	101	1	151	50
24.17	100	97	-3	147	50

24.33	98	89	-9	139	50
24.50	100	112	12	162	50
24.67	99	99	0	149	50
24.83	99	99	0	150	51
25.00	100	100	0	151	50
25.17	100	100	0	150	50
25.33	100	100	0	151	50
25.50	100	100	0	150	50
25.67	100	100	0	150	50
25.83	100	100	0	151	51
26.00	100	100	0	151	50
26.17	100	91	-9	141	50
26.33	99	108	9	158	50
26.50	100	99	-1	150	51
26.67	99	94	-5	143	49
26.83	100	106	6	157	51
27.00	100	100	0	151	50
27.17	100	100	0	151	50
27.33	100	100	0	151	51
27.50	100	100	0	150	51
27.67	100	100	0	150	50
27.83	100	100	0	150	50
28.00	100	100	0	150	50
28.17	100	100	0	150	50
28.33	100	100	0	151	51
28.50	100	100	0	150	50
28.67	100	100	0	150	50
28.83	100	100	0	150	50
29.00	100	100	-1	149	50
29.17	100	101	1	152	51
29.33	98	92	-6	142	50
29.50	100	105	5	155	50
29.67	100	101	1	151	50
29.83	100	100	0	151	51
30.00	100	100	0	151	50

Appendix B - Output for Reliable One Sender-to-Two Receivers Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets
0.00	0.90	0	-1	44.1	44
0.17	85.90	0	-86	49.9	50
0.33	99.90	0	-100	50.1	50
0.50	100.00	0	-100	50.3	50
0.67	100.10	0	-100	50.2	50
0.83	100.10	115	15	164	49
1.00	100.10	148	48	197.8	50
1.17	100.10	158	58	207.9	50
1.33	100.00	127	27	178.2	52
1.50	100.10	156	55	205.2	50
1.67	100.10	143	43	193.8	51
1.83	100.10	156	56	206	50
2.00	100.10	146	46	197.2	51
2.17	100.10	140	40	190.5	50
2.33	100.10	103	3	152.2	50
2.50	100.10	100	0	150.6	51
2.67	100.10	100	0	150.3	50
2.83	100.10	100	0	150.7	51
3.00	100.10	100	-1	150	50
3.17	100.10	101	1	150.5	50
3.33	100.10	100	-1	150.1	51
3.50	100.10	100	-1	149.9	50
3.67	98.80	100	1	149.3	50
3.83	99.90	100	0	150.5	51
4.00	100.10	100	0	150.3	50
4.17	100.10	100	0	150.4	50
4.33	100.10	100	0	150.5	50
4.50	100.10	100	0	150.2	50
4.67	100.10	100	0	150.4	50
4.83	100.10	100	0	150.4	50
5.00	100.10	100	0	150.5	50
5.17	100.00	100	0	150.5	51
5.33	100.10	100	0	150.2	50
5.50	100.10	98	-2	148.5	51
5.67	100.10	102	2	152.5	50
5.83	100.10	100	0	150.5	50
6.00	100.10	100	0	150.4	50
6.17	100.10	100	0	150.3	50
6.33	100.10	100	0	150.3	50
6.50	100.10	100	0	150.2	50
6.67	100.10	100	0	150.3	51
6.83	100.10	100	0	150.2	50
7.00	100.10	101	1	150.5	50
7.17	100.10	95	-5	146	51
7.33	100.10	105	5	154.9	50
7.50	100.10	100	0	150.4	51

7.67	100.00	100	0	150.5	50
7.83	100.10	77	-23	126.9	50
8.00	100.10	123	23	173.4	51
8.17	100.10	100	0	150.7	51
8.33	100.10	100	-1	149.2	50
8.50	100.10	98	-2	148.9	51
8.67	100.10	102	2	153.4	51
8.83	100.10	97	-3	147.7	51
9.00	100.10	103	3	153	50
9.17	100.10	100	0	150.3	50
9.33	100.10	100	0	150.4	50
9.50	100.10	100	0	149.7	50
9.67	100.10	100	0	151	51
9.83	100.10	100	0	150.4	50
10.00	100.10	100	0	150.3	50
10.17	100.10	100	0	150.5	50
10.33	99.90	100	0	150	50
10.50	100.10	100	0	150.3	50
10.67	100.00	100	0	150.4	51
10.83	99.20	99	0	149	50
11.00	100.00	100	0	150.3	50
11.17	100.10	100	0	150.3	50
11.33	100.10	100	0	150.7	51
11.50	100.10	100	0	150.4	50
11.67	100.10	100	0	150.4	50
11.83	100.10	100	0	150.4	50
12.00	100.10	100	0	150.4	50
12.17	100.10	100	0	150.4	51
12.33	99.70	100	0	150	50
12.50	100.10	100	0	150.4	50
12.67	100.10	100	0	149.7	50
12.83	100.10	100	0	150.9	51
13.00	100.10	100	0	150	50
13.17	100.10	97	-4	147.8	51
13.33	100.10	104	3	153.1	50
13.50	100.00	100	0	150.2	50
13.67	100.00	100	0	150.5	51
13.83	100.10	100	0	150.4	50
14.00	100.00	100	0	150.8	51
14.17	100.10	100	0	150.2	50
14.33	100.00	100	0	150.4	50
14.50	100.10	100	0	149.7	50
14.67	100.10	100	0	151.1	51
14.83	100.10	100	0	150.6	51
15.00	100.10	100	0	150.2	50
15.17	100.10	100	0	150.5	50
15.33	100.10	100	0	150.7	51
15.50	100.10	100	0	150.5	50
15.67	100.10	100	0	150.3	50
15.83	100.00	100	0	150.6	51

16.00	100.10	100	0	150.1	50
16.17	100.10	100	0	150.4	50
16.33	98.80	99	0	148.5	50
16.50	100.00	100	0	150.6	51
16.67	100.10	100	0	150.2	50
16.83	100.10	100	0	150.5	51
17.00	100.00	100	0	150.2	50
17.17	100.10	100	0	151	51
17.33	100.10	100	0	150.4	50
17.50	100.10	100	0	150.4	50
17.67	100.00	100	0	150.3	50
17.83	100.10	100	0	150.3	50
18.00	100.10	100	0	150.5	50
18.17	100.00	100	0	150.8	51
18.33	100.10	96	-4	146.5	51
18.50	100.10	104	4	154	50
18.67	100.10	100	0	150.3	50
18.83	100.00	100	0	150.1	50
19.00	100.10	100	0	150.9	51
19.17	100.10	90	-10	139.7	49
19.33	100.10	110	10	161.2	52
19.50	100.10	100	0	150.1	50
19.67	100.10	100	0	150.1	50
19.83	100.00	100	0	150.2	50
20.00	100.10	93	-7	143.8	51
20.17	100.10	100	0	150	50
20.33	99.90	105	5	155.2	50
20.50	100.00	89	-11	139.4	51
20.67	100.10	109	8	158.6	50
20.83	100.10	105	5	155.3	50
21.00	100.10	100	0	150.7	51
21.17	100.10	77	-23	127.9	51
21.33	100.10	121	21	170.5	50
21.50	100.10	102	2	152.5	50
21.67	100.10	100	0	150.2	50
21.83	100.10	100	0	150.2	50
22.00	100.10	100	0	150.2	50
22.17	100.10	96	-4	146.7	51
22.33	99.60	101	1	151	50
22.50	98.60	101	2	150.9	50
22.67	100.10	100	0	150	50
22.83	100.10	95	-5	145.3	51
23.00	100.10	105	5	155.1	50
23.17	100.00	76	-25	126.3	51
23.33	100.20	62	-38	112	50
23.50	100.10	117	17	166.6	49
23.67	100.00	137	37	187.1	51
23.83	100.00	102	2	152.7	51
24.00	100.10	108	8	157.6	50
24.17	100.00	88	-12	138.2	50

24.33	99.60	100	1	151.1	51
24.50	100.10	111	11	161.1	50
24.67	100.10	100	-1	150.1	51
24.83	100.10	101	1	150.4	50
25.00	100.00	99	-1	150.5	51
25.17	100.10	101	1	150.6	50
25.33	100.10	99	-1	149.3	50
25.50	100.10	91	-9	141.4	50
25.67	100.10	110	10	160.8	51
25.83	100.10	100	0	150.5	50
26.00	100.10	100	0	149.9	50
26.17	100.10	100	0	151	51
26.33	100.10	99	-1	149.8	51
26.50	99.90	97	-3	147.3	51
26.67	100.10	104	4	154.2	50
26.83	100.10	100	0	150.6	51
27.00	100.10	99	-1	149.1	50
27.17	100.00	101	1	151.1	50
27.33	100.10	100	0	150.6	51
27.50	100.10	100	0	150.4	50
27.67	100.10	100	0	150.3	50
27.83	100.10	100	0	150.2	50
28.00	100.00	100	0	150.5	51
28.17	98.00	93	-5	143	50
28.33	99.90	104	4	153.6	50
28.50	100.00	98	-2	149.1	51
28.67	100.10	100	0	148.7	49
28.83	100.00	103	3	154.7	52
29.00	100.10	96	-4	146.5	50
29.17	100.10	105	5	154.6	50
29.33	100.10	98	-2	147.8	50
29.50	100.10	102	2	153.2	51
29.67	100.10	100	0	150.5	50
29.83	100.10	100	0	150.3	50
30.00	100.10	100	0	150.5	51

Appendix C - Output for Reliable Two Sender-to-Two Receiver Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets
0.00	22.1	13.9	-8.2	27.4	13.5
0.17	94.8	64.8	-30	111.7	46.9
0.33	98.8	68.4	-30.4	117.6	49.2
0.50	98.6	65.3	-33.3	116	50.7
0.67	99.1	68.9	-30.2	119	50.1
0.83	99.7	68.7	-31	118.8	50.1
1.00	100.1	66.7	-33.4	118.2	51.5
1.17	99.8	61.2	-38.6	110.7	49.5
1.33	99.7	71.9	-27.8	124	52.1
1.50	99.6	70.3	-29.3	120.3	50
1.67	94.8	66	-28.8	114.2	48.2
1.83	96	73.1	-22.9	122.5	49.4
2.00	96.7	70	-26.7	118.4	48.4
2.17	95.6	63.2	-32.4	112.2	49
2.33	98.8	69.1	-29.7	119.7	50.6
2.50	99.4	71.4	-28	122.4	51
2.67	100	73.1	-26.9	123.7	50.6
2.83	99.8	65.2	-34.6	116.3	51.1
3.00	99.9	77.4	-22.5	127.8	50.4
3.17	99.7	65.9	-33.8	116.2	50.3
3.33	100	66.8	-33.2	117.2	50.4
3.50	99.9	75.7	-24.2	126.7	51
3.67	100	69.7	-30.3	121	51.3
3.83	100	70.8	-29.2	121.2	50.4
4.00	100.1	80.5	-19.6	131.3	50.8
4.17	98.5	70.4	-28.1	120.4	50
4.33	100	76.4	-23.6	127	50.6
4.50	100	70	-30	120.4	50.4
4.67	100.1	71	-29.1	122.1	51.1
4.83	100.1	75.1	-25	125.6	50.5
5.00	100.1	61.7	-38.4	112.5	50.8
5.17	99.9	52.6	-47.3	105	52.4
5.33	99.9	61.4	-38.5	110.3	48.9
5.50	99.9	70.8	-29.1	121.2	50.4
5.67	100	70.7	-29.3	121.5	50.8
5.83	99.9	63.9	-36	115.4	51.5
6.00	99.1	70	-29.1	120.1	50.1
6.17	95	60	-35	108.5	48.5
6.33	95.8	65.2	-30.6	113.3	48.1
6.50	96.3	71.4	-24.9	120.6	49.2
6.67	94	59.3	-34.7	107.1	47.8
6.83	99.8	62	-37.8	111.7	49.7
7.00	100	71.5	-28.5	122.2	50.7
7.17	100.1	58.6	-41.5	109.8	51.2
7.33	99.7	75	-24.7	124.3	49.3

7.50	100.1	70.6	-29.5	123.6	53
7.67	100.1	34.7	-65.4	83.4	48.7
7.83	99.7	75.1	-24.6	125.4	50.3
8.00	99.7	75	-24.7	125.9	50.9
8.17	100.1	67.6	-32.5	117.7	50.1
8.33	99.7	71.4	-28.3	122.5	51.1
8.50	100	35.7	-64.3	85.8	50.1
8.67	99.8	67.1	-32.7	118.7	51.6
8.83	100	77	-23	126.2	49.2
9.00	97	61.8	-35.2	110.8	49
9.17	96	64.4	-31.6	115.3	50.9
9.33	96.4	35.2	-61.2	82.8	47.6
9.50	95	65.7	-29.3	114.7	49
9.67	95.3	68.8	-26.5	117	48.2
9.83	99.8	67.1	-32.7	116.8	49.7
10.00	99.7	64.6	-35.1	116	51.4
10.17	99.5	67	-32.5	117.3	50.3
10.33	100	69.4	-30.6	118.9	49.5
10.50	100.1	72.3	-27.8	124	51.7
10.67	99.9	79.6	-20.3	130.8	51.2
10.83	100.1	67.9	-32.2	118.6	50.7
11.00	100.1	72.3	-27.8	123	50.7
11.17	100.1	58.3	-41.8	109.9	51.6
11.33	100	54	-46	103.8	49.8
11.50	100	74.3	-25.7	125	50.7
11.67	99.2	65.4	-33.8	116.3	50.9
11.83	95.9	66.8	-29.1	114.7	47.9
12.00	96	67.5	-28.5	116.6	49.1
12.17	96	64.3	-31.7	114.5	50.2
12.33	94.2	60.9	-33.3	108.8	47.9
12.50	99.7	71	-28.7	121.3	50.3
12.67	100	69.2	-30.8	120.3	51.1
12.83	99.9	68.5	-31.4	118.9	50.4
13.00	100	72.1	-27.9	123.5	51.4
13.17	99.9	60.6	-39.3	111.2	50.6
13.33	100.1	68.6	-31.5	118.5	49.9
13.50	100.1	70.1	-30	120.4	50.3
13.67	100	73.1	-26.9	123.5	50.4
13.83	99.9	66.9	-33	118.1	51.2
14.00	100.1	70.9	-29.2	121.3	50.4
14.17	99.1	70.8	-28.3	122.4	51.6
14.33	100	69.5	-30.5	119.4	49.9
14.50	99.6	68.4	-31.2	119.5	51.1
14.67	99.9	74.1	-25.8	124.7	50.6
14.83	99.8	74.7	-25.1	125.8	51.1
15.00	100.1	70.9	-29.2	121.1	50.2
15.17	99.7	71.8	-27.9	122.3	50.5
15.33	96.2	72.3	-23.9	121.1	48.8
15.50	95	64.5	-30.5	113.5	49
15.67	95.5	68.8	-26.7	116.7	47.9

15.83	95.5	65.3	-30.2	114.6	49.3
16.00	96.8	64.7	-32.1	113.7	49
16.17	95.8	72.1	-23.7	120.6	48.5
16.33	99.9	75.3	-24.6	125.7	50.4
16.50	100	63.5	-36.5	114.3	50.8
16.67	100.1	68.7	-31.4	119.4	50.7
16.83	100.1	69.8	-30.3	119.7	49.9
17.00	99.9	77.2	-22.7	128.5	51.3
17.17	99.9	49.8	-50.1	100.3	50.5
17.33	99.5	65.6	-33.9	116.6	51
17.50	99.7	71.8	-27.9	121.8	50
17.67	99.9	72.8	-27.1	124.3	51.5
17.83	100.1	73.8	-26.3	124.4	50.6
18.00	100	74.7	-25.3	125	50.3
18.17	99.8	69.8	-30	121.1	51.3
18.33	97.9	77.2	-20.7	126.1	48.9
18.50	99.9	72.8	-27.1	124.6	51.8
18.67	99.7	72	-27.7	122.7	50.7
18.83	100.1	71.5	-28.6	121.9	50.4
19.00	100.1	70.2	-29.9	120.6	50.4
19.17	99.8	70.3	-29.5	121.8	51.5
19.33	100	70.1	-29.9	119.9	49.8
19.50	99.5	77.4	-22.1	127.9	50.5
19.67	99.4	69.4	-30	120	50.6
19.83	99.5	70.2	-29.3	120.8	50.6
20.00	100.1	70.3	-29.8	120.9	50.6
20.17	100.1	71.7	-28.4	122.8	51.1
20.33	99.9	64.6	-35.3	115.6	51
20.50	99.9	60.8	-39.1	111.8	51
20.67	99.7	79.4	-20.3	129.9	50.5
20.83	100.1	71.5	-28.6	122.2	50.7
21.00	99.9	67	-32.9	118.2	51.2
21.17	99.8	67.8	-32	118.3	50.5
21.33	98.1	72	-26.1	121.8	49.8
21.50	100.1	73.3	-26.8	124.5	51.2
21.67	100.1	66.1	-34	116.9	50.8
21.83	99.8	70.3	-29.5	120.2	49.9
22.00	100.1	71.6	-28.5	121.6	50
22.17	99	75.1	-23.9	126.8	51.7
22.33	99.9	65.3	-34.6	114.9	49.6
22.50	100	70.9	-29.1	121.9	51
22.67	100	76.3	-23.7	126.7	50.4
22.83	99.8	69.1	-30.7	120.6	51.5
23.00	100	69.1	-30.9	118.9	49.8
23.17	99.9	78.8	-21.1	130.7	51.9
23.33	100.1	62.3	-37.8	112.2	49.9
23.50	100	70.9	-29.1	120.4	49.5
23.67	99.9	72.3	-27.6	124.1	51.8
23.83	100.1	73.7	-26.4	124.4	50.7
24.00	100	71.1	-28.9	121.5	50.4

24.17	99.5	72.6	-26.9	123.1	50.5
24.33	100	71.3	-28.7	121.3	50
24.50	100	67.5	-32.5	117.8	50.3
24.67	99.8	76.7	-23.1	128.4	51.7
24.83	100	68.6	-31.4	119.7	51.1
25.00	99.8	69	-30.8	118.9	49.9
25.17	97.5	67.7	-29.8	117.4	49.7
25.33	100	70.6	-29.4	121.2	50.6
25.50	100	74.3	-25.7	125	50.7
25.67	100.1	69.5	-30.6	120.2	50.7
25.83	100	67.4	-32.6	118.1	50.7
26.00	100	76	-24	127.3	51.3
26.17	99.5	67.9	-31.6	118.6	50.7
26.33	100.1	66.5	-33.6	117.1	50.6
26.50	100.1	81.5	-18.6	132	50.5
26.67	99.9	69.1	-30.8	119.9	50.8
26.83	100.1	73.5	-26.6	125	51.5
27.00	100.1	71.6	-28.5	121.2	49.6
27.17	99.8	67.1	-32.7	117.6	50.5
27.33	100	62.7	-37.3	113.2	50.5
27.50	100.1	69	-31.1	119.4	50.4
27.67	99.7	72	-27.7	122.6	50.6
27.83	99.8	74.5	-25.3	125.1	50.6
28.00	100.1	72.3	-27.8	123.4	51.1
28.17	100	77	-23	128.8	51.8
28.33	100.1	70.6	-29.5	120.2	49.6
28.50	100	73	-27	123.7	50.7
28.67	100	69.6	-30.4	121	51.4
28.83	100	75.4	-24.6	125.9	50.5
29.00	100.1	71.3	-28.8	122.1	50.8
29.17	100.1	58.6	-41.5	109.2	50.6
29.33	99.5	74.3	-25.2	124.8	50.5
29.50	100	69.1	-30.9	119.8	50.7
29.67	100	67.7	-32.3	118	50.3
29.83	99.7	73.8	-25.9	124	50.2
30.00	100	69.9	-30.1	121.5	51.6

Appendix D - Output for Unreliable One Sender-to-One Receivers Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets	Drop Packets
0.17	1	1	0	38	37	0
0.33	82	89	7	132	44	0
0.50	96	86	-10	139	53	4
0.67	100	102	2	152	50	10
0.83	100	105	5	155	50	10
1.00	94	69	-25	117	48	10
1.17	99	105	6	155	50	10
1.33	99	83	-16	132	50	10
1.50	98	121	23	170	49	10
1.67	98	110	12	159	50	10
1.83	99	99	1	149	50	10
2.00	98	96	-2	145	49	10
2.17	93	94	2	141	47	10
2.33	91	89	-3	135	47	10
2.50	97	100	3	150	49	10
2.67	99	90	-8	140	49	10
2.83	99	105	6	154	49	10
3.00	97	99	2	148	49	10
3.17	98	98	0	148	50	10
3.33	99	99	0	148	50	10
3.50	99	100	0	150	50	10
3.67	98	98	0	147	50	10
3.83	99	99	0	149	50	10
4.00	98	96	-2	144	49	10
4.17	90	92	2	140	48	10
4.33	98	97	-1	146	49	10
4.50	99	101	1	151	50	10
4.67	100	100	0	150	50	10
4.83	100	95	-5	145	50	10
5.00	100	100	0	150	50	10
5.17	100	104	4	153	50	10
5.33	90	87	-3	134	47	10
5.50	99	103	4	153	50	10
5.67	100	96	-4	147	51	10
5.83	100	82	-18	133	51	10
6.00	100	98	-3	148	50	10
6.17	100	108	8	159	50	10
6.33	100	88	-12	139	51	10
6.50	96	119	23	168	48	10
6.67	96	101	5	151	50	10
6.83	100	101	1	151	50	10
7.00	100	100	0	150	50	10
7.17	100	101	1	151	50	10
7.33	93	93	0	140	48	10
7.50	98	98	0	148	50	10

7.67	100	100	0	150	50	10
7.83	92	91	-1	139	47	10
8.00	92	93	2	140	47	10
8.17	100	100	0	150	50	10
8.33	100	99	-1	150	51	10
8.50	100	97	-3	147	50	10
8.67	95	99	4	148	49	10
8.83	100	97	-3	147	50	10
9.00	99	102	3	152	50	10
9.17	100	98	-3	148	50	10
9.33	100	103	3	153	50	10
9.50	100	100	0	150	50	10
9.67	100	100	0	150	50	10
9.83	100	100	0	151	50	10
10.00	92	88	-4	135	47	10
10.17	99	102	4	152	50	10
10.33	99	100	1	150	50	10
10.50	100	100	-1	150	51	10
10.67	100	101	1	151	50	10
10.83	100	88	-12	139	50	10
11.00	100	107	7	158	50	10
11.17	100	105	5	155	50	10
11.33	93	92	-1	139	48	10
11.50	100	101	1	151	50	10
11.67	99	99	0	149	50	10
11.83	100	100	0	151	50	10
12.00	100	100	0	150	50	10
12.17	100	100	0	151	50	10
12.33	100	100	0	150	50	10
12.50	100	97	-4	147	50	10
12.67	100	104	4	154	50	10
12.83	100	99	-1	150	51	10
13.00	100	96	-4	147	51	10
13.17	100	105	5	155	50	10
13.33	100	91	-9	141	51	10
13.50	100	102	2	153	51	10
13.67	100	93	-8	143	51	10
13.83	100	114	14	165	50	10
14.00	100	102	2	152	50	10
14.17	100	96	-4	147	50	10
14.33	100	82	-18	132	50	10
14.50	100	120	20	170	50	10
14.67	99	100	2	150	50	10
14.83	100	98	-2	148	50	10
15.00	93	79	-14	127	48	10
15.17	100	87	-12	138	50	10
15.33	93	90	-3	139	49	10
15.50	100	110	10	160	50	10
15.67	100	106	7	157	50	10
15.83	100	97	-3	147	50	10

16.00	100	116	16	167	50	10
16.17	100	102	2	153	50	10
16.33	100	100	0	151	50	10
16.50	100	100	0	150	50	10
16.67	100	99	-1	149	50	10
16.83	100	101	1	151	50	10
17.00	100	101	1	151	50	10
17.17	100	100	0	150	50	10
17.33	100	100	-1	150	50	10
17.50	100	95	-5	145	51	10
17.67	100	89	-11	139	50	10
17.83	100	113	13	163	50	10
18.00	100	104	4	155	51	10
18.17	99	99	0	148	50	10
18.33	100	100	0	151	50	10
18.50	100	100	0	150	50	10
18.67	100	100	0	150	50	10
18.83	100	100	-1	150	51	10
19.00	100	101	1	151	50	10
19.17	100	100	0	151	50	10
19.33	100	99	-1	149	50	10
19.50	100	99	-1	150	51	10
19.67	100	102	1	152	50	10
19.83	100	100	-1	150	51	10
20.00	100	100	0	151	50	10
20.17	100	101	1	151	50	10
20.33	100	100	0	151	51	10
20.50	100	100	0	150	50	10
20.67	100	100	0	151	51	10
20.83	100	100	0	150	50	10
21.00	100	100	0	151	50	10
21.17	100	100	0	150	50	10
21.33	100	100	0	151	50	10
21.50	100	100	0	150	50	10
21.67	99	100	0	150	50	10
21.83	100	100	0	150	50	10
22.00	100	100	0	151	51	10
22.17	100	100	0	150	50	10
22.33	100	100	0	150	50	10
22.50	100	92	-9	142	50	10
22.67	99	108	9	157	49	10
22.83	100	101	1	151	51	10
23.00	100	100	0	150	50	10
23.17	100	100	0	150	50	10
23.33	100	100	0	151	50	10
23.50	100	100	0	150	50	10
23.67	100	100	0	150	50	10
23.83	100	100	0	150	50	10
24.00	100	98	-2	148	50	10
24.17	92	94	2	141	47	10

24.33	100	100	0	151	50	10
24.50	100	87	-13	137	50	10
24.67	95	94	-1	143	49	10
24.83	100	112	12	163	50	10
25.00	100	82	-18	132	50	10
25.17	100	99	0	150	50	10
25.33	100	118	18	169	50	10
25.50	100	102	2	152	50	10
25.67	100	100	0	150	50	10
25.83	100	78	-23	128	50	10
26.00	100	103	3	154	51	10
26.17	99	118	19	168	50	10
26.33	100	100	0	151	51	10
26.50	100	100	0	151	51	10
26.67	100	100	0	151	50	10
26.83	100	100	0	150	50	10
27.00	100	100	0	150	50	10
27.17	100	90	-10	140	50	10
27.33	100	97	-3	147	50	10
27.50	100	113	13	164	50	10
27.67	100	100	0	150	50	10
27.83	100	97	-3	148	50	10
28.00	100	91	-9	141	50	10
28.17	100	110	10	160	50	10
28.33	100	97	-3	147	50	10
28.50	99	105	5	154	50	10
28.67	100	100	0	150	50	10
28.83	100	100	0	150	50	10
29.00	95	95	0	144	48	10
29.17	100	100	0	150	50	10
29.33	100	101	1	151	50	10
29.50	100	100	0	150	50	10
29.67	100	100	0	151	51	10
29.83	100	98	-2	148	50	10
30.00	100	88	-12	138	50	10

Appendix E - Output for Unreliable One Sender-to-Two Receivers Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets	Drop Packets
0.17	99.60	85	-15	134.8	50	10
0.33	98.60	100	1	149.4	50	10.1
0.50	99.00	110	11	159.6	50	9.8
0.67	98.50	80	-18	130.1	50	10.2
0.83	98.60	88	-11	137.4	50	9.9
1.00	98.60	62	-37	111.3	50	10
1.17	98.90	67	-32	117.3	50	10
1.33	97.80	107	9	156.3	50	10
1.50	100.00	113	13	163.6	50	10.1
1.67	100.10	102	2	151.9	50	9.9
1.83	100.10	66	-34	116.4	50	10
2.00	100.10	102	2	152.4	50	10
2.17	100.10	109	8	158.8	50	10.1
2.33	100.10	103	3	153.1	50	9.9
2.50	100.10	59	-41	109.7	50	10
2.67	100.10	41	-59	91.3	50	10
2.83	100.00	109	9	159.6	50	10
3.00	97.00	107	10	156	49	10
3.17	99.60	70	-30	120.3	50	10
3.33	98.50	102	4	151.7	50	10
3.50	98.40	93	-6	142.3	50	10.1
3.67	98.10	116	17	164.7	49	10
3.83	97.80	80	-17	129.6	49	9.9
4.00	99.10	111	12	160.9	50	10
4.17	99.10	109	10	158.7	50	10.1
4.33	98.20	86	-13	135.3	50	9.9
4.50	98.50	109	11	158.9	50	10.1
4.67	94.60	109	14	157.3	49	9.9
4.83	99.10	103	4	153	50	10
5.00	99.10	90	-9	139.9	50	10
5.17	100.10	116	16	166.2	50	10
5.33	100.00	82	-18	132.7	50	10.1
5.50	100.00	108	8	158.5	51	10
5.67	100.10	123	23	173.5	51	9.9
5.83	100.10	112	11	161.9	50	10
6.00	99.80	120	20	170.3	50	10
6.17	100.10	109	8	158.8	50	10
6.33	100.10	96	-5	145.7	50	10.1
6.50	100.10	118	18	167.8	50	9.9
6.67	100.00	65	-35	115.2	50	10.1
6.83	99.90	99	-1	149.5	50	9.9
7.00	100.00	98	-2	148.2	51	10
7.17	99.90	118	18	168.1	50	10
7.33	100.10	114	14	164.3	51	10.1
7.50	100.10	122	22	172.3	51	10

7.67	97.80	107	9	155.7	49	9.9
7.83	99.90	115	15	164.8	50	10
8.00	100.10	98	-2	147.9	50	10.1
8.17	100.10	124	24	173.9	50	9.9
8.33	99.90	122	22	172	50	10.1
8.50	100.10	106	6	156.1	50	10
8.67	100.00	95	-5	145.3	50	9.9
8.83	100.10	111	10	160.8	50	10
9.00	100.00	112	12	162.2	50	10
9.17	99.80	102	2	151.9	50	10.1
9.33	100.10	119	19	169.6	50	9.9
9.50	99.90	120	20	169.7	50	9.9
9.67	99.90	110	11	160.6	50	10.1
9.83	99.90	119	19	169	50	9.9
10.00	99.50	113	14	163.4	50	10.1
10.17	100.10	113	13	163.6	50	10
10.33	100.10	136	36	186.8	50	9.9
10.50	100.00	123	23	173.5	50	10.2
10.67	100.00	122	22	172.2	50	9.9
10.83	100.10	94	-6	144.3	50	9.9
11.00	100.10	82	-18	132.3	51	10.1
11.17	100.10	108	8	158.9	51	9.9
11.33	100.10	120	20	169.9	50	10.2
11.50	100.10	113	13	163.4	50	9.9
11.67	100.10	114	14	164.3	50	10
11.83	99.80	98	-2	148.2	50	10
12.00	100.00	118	18	168	50	10.1
12.17	100.10	113	13	163.1	50	9.9
12.33	100.10	59	-41	109.2	50	9.9
12.50	100.10	101	1	151.4	50	10.1
12.67	100.00	92	-8	142.2	50	10
12.83	99.50	91	-9	141.1	50	9.9
13.00	99.60	120	20	170.2	50	10.1
13.17	99.50	110	10	160.3	50	10.1
13.33	100.00	108	8	157.9	50	10
13.50	100.00	119	19	169	50	9.9
13.67	100.10	118	18	168	50	10.1
13.83	100.10	116	16	165.9	50	9.9
14.00	100.10	124	24	174.5	50	10
14.17	100.10	87	-13	137.2	50	10
14.33	100.10	103	3	153.3	50	10
14.50	100.10	60	-41	110	50	9.9
14.67	98.30	75	-23	124.7	49	10.1
14.83	100.10	98	-3	148	50	10.1
15.00	100.10	107	7	157.5	50	10
15.17	100.10	118	18	168.2	51	10
15.33	100.10	71	-29	121.4	50	9.8
15.50	99.90	64	-36	114.7	51	10.1
15.67	100.00	102	2	152.1	50	10
15.83	100.00	102	2	152.9	51	10

16.00	99.80	104	4	154.3	50	10
16.17	100.10	107	7	157.1	50	9.9
16.33	100.00	98	-2	148.6	50	10.1
16.50	100.10	97	-4	146.7	50	10
16.67	100.10	115	15	165.8	50	10
16.83	100.10	101	1	150.9	50	10
17.00	100.10	89	-12	138.8	50	10
17.17	100.10	76	-24	126.1	50	9.9
17.33	100.10	50	-50	100.4	50	10.1
17.50	99.20	115	16	165.4	50	10
17.67	100.00	104	4	154	51	10
17.83	99.90	72	-28	122.2	50	10
18.00	100.00	56	-44	106	50	9.9
18.17	100.10	109	9	159.5	51	10.1
18.33	100.10	101	1	151.8	51	10
18.50	100.10	114	14	163.8	50	10
18.67	100.10	92	-8	142.1	50	10
18.83	100.00	79	-21	129.1	50	9.9
19.00	100.10	106	6	156.4	51	10.2
19.17	100.10	95	-6	144.9	50	9.8
19.33	100.10	109	9	159.1	50	10.1
19.50	100.10	96	-4	146.2	50	10.1
19.67	100.10	116	16	166.6	50	10
19.83	100.00	86	-14	136.1	51	9.9
20.00	100.10	66	-34	116.6	50	10
20.17	99.70	102	3	152.7	51	10.1
20.33	100.10	84	-17	133.7	50	10
20.50	100.00	109	9	159.2	50	9.8
20.67	100.10	99	-2	149.1	51	10.1
20.83	100.00	104	4	154.8	51	10
21.00	100.10	87	-14	137	50	10
21.17	100.10	112	12	162.6	50	10.1
21.33	100.10	115	15	164.8	50	9.9
21.50	100.10	111	11	161.5	50	10.1
21.67	98.00	107	9	156.4	49	9.9
21.83	100.10	117	17	167.3	50	10
22.00	99.40	114	14	164.5	51	10.1
22.17	100.10	101	1	151.5	50	10
22.33	100.10	102	2	152.4	51	9.9
22.50	100.10	112	12	162.5	50	10
22.67	100.10	105	5	155.4	50	10.1
22.83	100.10	67	-34	116.7	50	9.9
23.00	100.10	115	15	165.2	50	10
23.17	100.10	87	-14	136.7	50	10.1
23.33	99.90	113	13	163.3	50	9.8
23.50	100.10	111	11	161.7	50	10.1
23.67	100.00	114	14	164.5	50	10
23.83	100.00	96	-4	145.7	50	9.9
24.00	100.00	120	20	170	50	10.2
24.17	100.00	112	12	162.1	50	9.9

24.33	100.10	119	19	168.8	50	10
24.50	100.10	128	28	178.7	51	10
24.67	100.10	118	18	168.2	50	10
24.83	100.00	99	-2	148.8	50	10
25.00	99.90	64	-36	114	50	9.9
25.17	100.10	98	-3	148.1	51	10.2
25.33	100.10	115	15	165.1	51	9.9
25.50	100.00	120	20	170.6	51	10.1
25.67	100.10	121	21	171.6	50	9.9
25.83	100.00	126	26	176.7	50	10
26.00	100.10	106	6	156.7	50	10.1
26.17	100.10	117	17	167	50	10
26.33	100.10	110	10	159.9	50	10
26.50	100.00	117	17	167.4	50	9.9
26.67	100.00	122	22	172.1	50	10
26.83	100.00	91	-9	140.8	50	10
27.00	100.10	119	18	168.5	50	10
27.17	100.10	113	13	162.9	50	10
27.33	100.00	125	25	175	51	10
27.50	99.20	126	27	176.6	50	10
27.67	100.10	118	18	168.2	50	10
27.83	99.70	126	26	176.4	50	10.1
28.00	100.10	99	-1	149.7	50	9.8
28.17	100.00	119	19	169.2	50	10.1
28.33	100.10	114	14	164.6	50	10.1
28.50	100.10	97	-4	147	50	9.9
28.67	98.40	111	12	160.2	50	9.9
28.83	100.10	106	6	156.7	50	10.1
29.00	100.00	123	23	173.5	50	10.1
29.17	100.10	126	26	176.1	50	9.9
29.33	99.90	107	7	157	50	9.9
29.50	100.10	129	29	178.9	50	10.1
29.67	100.10	87	-13	137.1	50	10
29.83	100.10	110	10	160.1	50	10.1
30.00	100.00	111	11	160.9	50	10
30.17	100.00	89	-11	140	51	9.9
30.33	100.00	103	3	153.3	50	10

Appendix F - Output for Unreliable Two Sender-to-Two Receivers Test Run

Time(Minutes)	App Messages	LSMP Packets	LSMP-App	Throughput Packets	Resent Packets
0.00	0.9	0.9	0	3.6	2.7
0.17	93.2	45.1	-48.1	93.1	48
0.33	98.8	59.8	-39	109.4	49.6
0.50	99.7	69.4	-30.3	120.4	51
0.67	99.7	79.2	-20.5	130.3	51.1
0.83	99.6	69.8	-29.8	120.9	51.1
1.00	98.5	69.3	-29.2	119.9	50.6
1.17	99.9	67.7	-32.2	118.7	51
1.33	99.9	69.5	-30.4	120.1	50.6
1.50	99.7	76.5	-23.2	127.5	51
1.67	97.9	70.5	-27.4	120.5	50
1.83	99.8	54.6	-45.2	106	51.4
2.00	99.6	59.6	-40	110.5	50.9
2.17	98.8	42.3	-56.5	92.8	50.5
2.33	99.7	51.6	-48.1	102.4	50.8
2.50	100	52.2	-47.8	103	50.8
2.67	100	54.3	-45.7	105.1	50.8
2.83	99.9	56	-43.9	106.7	50.7
3.00	100	58	-42	109.2	51.2
3.17	99.3	53.9	-45.4	104.6	50.7
3.33	98.9	58.3	-40.6	108.4	50.1
3.50	100	53.2	-46.8	104.4	51.2
3.67	99.9	56.9	-43	107.7	50.8
3.83	100	56.8	-43.2	107.5	50.7
4.00	99.8	62.2	-37.6	112.7	50.5
4.17	99.3	57.4	-41.9	108.3	50.9
4.33	98.5	60.4	-38.1	111.6	51.2
4.50	99.3	60.9	-38.4	111.2	50.3
4.67	96.5	54.1	-42.4	104.5	50.4
4.83	99.8	55.7	-44.1	106.1	50.4
5.00	99.2	49.1	-50.1	99.7	50.6
5.17	98.9	51.1	-47.8	101.7	50.6
5.33	99.2	56.6	-42.6	107.4	50.8
5.50	99.8	62.1	-37.7	112.8	50.7
5.67	98	49	-49	99.4	50.4
5.83	100	33.2	-66.8	83.6	50.4
6.00	100	53	-47	103.8	50.8
6.17	99.7	57.1	-42.6	107.8	50.7
6.33	99.8	54.4	-45.4	105.4	51
6.50	100	57.8	-42.2	108.8	51
6.67	99.9	53.4	-46.5	104	50.6
6.83	99.8	66	-33.8	116.9	50.9
7.00	99.8	55.9	-43.9	106.6	50.7
7.17	99.9	53.3	-46.6	104.3	51
7.33	99	43.4	-55.6	93.7	50.3

7.50	99	55.1	-43.9	105.4	50.3
7.67	98.8	60.7	-38.1	111	50.3
7.83	99.4	55.8	-43.6	106.4	50.6
8.00	98.7	46.7	-52	97.2	50.5
8.17	98.3	57.7	-40.6	108	50.3
8.33	99.5	59.6	-39.9	110.1	50.5
8.50	98.9	59.6	-39.3	109.7	50.1
8.67	98.5	53.9	-44.6	104.2	50.3
8.83	97.5	63.3	-34.2	113.3	50
9.00	99.8	60.7	-39.1	111.3	50.6
9.17	99.5	56.2	-43.3	106.9	50.7
9.33	100.1	56	-44.1	106.6	50.6
9.50	100.1	62.3	-37.8	113	50.7
9.67	99.9	57.9	-42	108.9	51
9.83	100.1	54.6	-45.5	105.1	50.5
10.00	100.1	60.1	-40	110.9	50.8
10.17	99.8	52.9	-46.9	103.7	50.8
10.33	99.8	61	-38.8	112	51
10.50	99.9	50.3	-49.6	100.9	50.6
10.67	99.4	59.1	-40.3	109.8	50.7
10.83	99.4	49.2	-50.2	99.5	50.3
11.00	98.9	43.4	-55.5	93.2	49.8
11.17	99.6	47.9	-51.7	98.4	50.5
11.33	99.1	55.3	-43.8	105.9	50.6
11.50	98.7	61.4	-37.3	111.9	50.5
11.67	99.3	59.2	-40.1	110.1	50.9
11.83	98.7	49.8	-48.9	99.9	50.1
12.00	97.8	57.3	-40.5	107	49.7
12.17	99.3	51.4	-47.9	101.8	50.4
12.33	100.1	46.7	-53.4	97.8	51.1
12.50	99.9	58.7	-41.2	110	51.3
12.67	99.8	52.9	-46.9	104	51.1
12.83	99.8	63.3	-36.5	113.9	50.6
13.00	100	59.9	-40.1	110.4	50.5
13.17	100	46.7	-53.3	97.2	50.5
13.33	99.9	55.2	-44.7	105.9	50.7
13.50	99.8	53.9	-45.9	104.5	50.6
13.67	100	61.2	-38.8	112.4	51.2
13.83	99.7	56.4	-43.3	107.5	51.1
14.00	98.7	59	-39.7	109.4	50.4
14.17	99.5	104.7	5.2	155.2	50.5
14.33	98.9	102.2	3.3	152.4	50.2
14.50	99	100.9	1.9	151	50.1
14.67	98.5	110.6	12.1	160.5	49.9
14.83	99.6	108.4	8.8	158.7	50.3
15.00	98.9	113.9	15	163.8	49.9
15.17	99.5	83.7	-15.8	133.9	50.2
15.33	98.9	105.4	6.5	155.6	50.2
15.50	99.6	106.4	6.8	156.9	50.5
15.67	99.5	106.7	7.2	156.8	50.1

15.83	99.1	107.7	8.6	157.8	50.1
16.00	97.4	102.6	5.2	152.3	49.7
16.17	97.9	103.2	5.3	152.9	49.7
16.33	98.9	113.1	14.2	163.4	50.3
16.50	98	72.1	-25.9	121.6	49.5
16.67	97.5	97	-0.5	146.5	49.5
16.83	97.9	111.9	14	161.5	49.6
17.00	98	109.3	11.3	158.7	49.4
17.17	98	87.1	-10.9	136.9	49.8
17.33	98.4	102.9	4.5	153	50.1
17.50	97.8	111.5	13.7	161.3	49.8
17.67	97.9	105.6	7.7	155.5	49.9
17.83	98	111.3	13.3	160.6	49.3
18.00	98.5	108.3	9.8	158	49.7
18.17	98.4	104.8	6.4	154.7	49.9
18.33	97.7	107.8	10.1	156.7	48.9
18.50	99.6	107.1	7.5	157.8	50.7
18.67	99.8	104	4.2	154.6	50.6
18.83	99.7	110.3	10.6	161.1	50.8
19.00	100	107	7	157.3	50.3
19.17	99.9	108.1	8.2	158.7	50.6
19.33	99.9	109	9.1	159.5	50.5
19.50	100	103.8	3.8	154.4	50.6
19.67	100	99.9	-0.1	150.5	50.6
19.83	100	105.4	5.4	156	50.6
20.00	99.9	114.2	14.3	165	50.8
20.17	99.9	107.3	7.4	157.7	50.4
20.33	100	106.8	6.8	157.7	50.9
20.50	100	112.4	12.4	162.9	50.5
20.67	100	103.8	3.8	154.2	50.4
20.83	100	101.9	1.9	152.7	50.8
21.00	100	101.3	1.3	152	50.7
21.17	100	102.4	2.4	153.2	50.8
21.33	100	104.2	4.2	154.7	50.5
21.50	99.8	80.6	-19.2	131	50.4
21.67	100	103.9	3.9	154.8	50.9
21.83	100	105.9	5.9	156.5	50.6
22.00	100	109.6	9.6	160.3	50.7
22.17	100	115.6	15.6	166.6	51
22.33	100	105.2	5.2	156	50.8
22.50	99.9	109.1	9.2	159.4	50.3
22.67	100	116.9	16.9	167.4	50.5
22.83	100	103.2	3.2	153.9	50.7
23.00	100	115.1	15.1	165.7	50.6
23.17	100	92.8	-7.2	143.5	50.7
23.33	99.9	117.2	17.3	167.8	50.6
23.50	100	109	9	159.5	50.5
23.67	100	120.1	20.1	171	50.9
23.83	100	115.4	15.4	166.1	50.7
24.00	100	107.9	7.9	158.4	50.5

24.17	99.3	107.1	7.8	158.1	51
24.33	99.9	109.2	9.3	159.9	50.7
24.50	99.9	106.5	6.6	157	50.5
24.67	99.9	112.6	12.7	163	50.4
24.83	98.5	91.9	-6.6	141.9	50
25.00	99.9	103.3	3.4	154	50.7
25.17	99.6	99.3	-0.3	149.7	50.4
25.33	100	78.5	-21.5	129.1	50.6
25.50	99.9	111	11.1	162.2	51.2
25.67	100	105.8	5.8	156.1	50.3
25.83	99.9	98.8	-1.1	149.4	50.6
26.00	100	103.6	3.6	154.4	50.8
26.17	98.2	97.5	-0.7	147	49.5
26.33	100	85.8	-14.2	136.7	50.9
26.50	100	107.6	7.6	158.4	50.8
26.67	100	113.7	13.7	164.2	50.5
26.83	100	117.4	17.4	168.1	50.7
27.00	100	108	8	158.8	50.8
27.17	99.9	86.8	-13.1	137.2	50.4
27.33	100	93.5	-6.5	144.2	50.7
27.50	100	81.4	-18.6	131.8	50.4
27.67	100	103.1	3.1	153.9	50.8
27.83	99.9	106.9	7	157.4	50.5
28.00	100	109.1	9.1	159.6	50.5
28.17	99	114.3	15.3	165.1	50.8
28.33	100	108.9	8.9	159.3	50.4
28.50	100	112	12	162.4	50.4
28.67	100	113.7	13.7	164.2	50.5
28.83	99.9	116	16.1	166.5	50.5
29.00	100	112.7	12.7	163.5	50.8
29.17	100	103.7	3.7	154.3	50.6
29.33	100	110.4	10.4	161.1	50.7
29.50	100	81.4	-18.6	132	50.6
29.67	100	112.1	12.1	162.4	50.3
29.83	100	110.9	10.9	161.5	50.6
30.00	100	102.7	2.7	153.1	50.4