

Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

2005

Framework for throughput analysis of simple reliable multicast protocol in a M2MP network

Romi Saluja

Follow this and additional works at: <https://repository.rit.edu/theses>

Recommended Citation

Saluja, Romi, "Framework for throughput analysis of simple reliable multicast protocol in a M2MP network" (2005). Thesis. Rochester Institute of Technology. Accessed from

This Master's Project is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

**Throughput Analysis of Simple Reliable Multicast Protocol in a
M2MP Network**

Masters Project Proposal

Romi Saluja

March 15th, 2004

Faculty Chairman: Professor Hans Peter Bischof

Faculty Reader: Professor Alan R Kaminsky

Table of Contents

TABLE OF CONTENTS	2
1.0 INTRODUCTION.....	3
2.0 PROJECT OBJECTIVES.....	3
2.1 DROP PACKET LAYER.....	3
2.2 THROUGHPUT LAYER.....	4
2.3 RELIABLE MULTICAST API.....	4
2.3.1 <i>General Design Considerations for a Multicast API</i>	4
2.3.1.1 Application Oriented.....	4
2.3.1.2 Sender or Receiver Oriented	5
2.3.1.3 Congestion Control	5
2.3.1.4 Network Topologies.....	6
2.3.1.5 Security, Authentication and Privacy	6
2.3.2 <i>Simple Reliable Multicast API</i>	6
2.4 APPLICATION	7
3.0 PROJECT ANALYSIS.....	7
4.0 PROJECT ARCHITECTURE	7
BIBLIOGRAPHY	9

1.0 Introduction

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 adhoc 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 adhoc 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].

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 adhoc wireless network of mobile devices, where devices are constantly joining and leaving the network. This Masters project aims to satisfy this requirement by developing a service, which would simulate an unreliable network.

2.0 Project Objectives

2.1 DropPacket Layer

The primary objective of this project is to test the Many to Many Protocol (M2MP) in a simulated unreliable network. In order to simulate an unreliable network, a DropPacket layer will be designed, developed and added to the M2M protocol stack as a channel implementation. It will be the responsibility of the DropPacket layer to prevent data packets handed down by the application to be broadcast on to the network. The packets that are prevented from being broadcast are analogous to the packets that would be dropped in an unreliable wireless network. The DropPacket layer will be able to

dynamically accept the input for the number of packets to be dropped. This would enable simulating a network environment with varying levels of unreliability. Also, the DropPacket layer will have the ability to monitor the data packet traffic between the applications i.e. the number of packets sent by the sender application and the number of packets received by the receiver application.

2.2 Throughput Layer

In addition to the DropPacket Service, a throughput-control layer would be integrated into the existing M2MP stack [10]. The throughput-control layer would enable simulating a wireless environment by providing an ability to control the rate of send and receive throughput.

2.3 Reliable Multicast API

As part of the project, a Simple Reliable Multicast API will be developed. An application will be developed to use the API to send and receive packets over the M2MP network

2.3.1 General Design Considerations for a Multicast API

Some of the general design considerations in development of a reliable multicast protocol are

2.3.1.1 Application Oriented

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 [2]. Some applications require fast detection of data loss [4] while others require time bound delivery of data [2]. 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 services embedded in the application and the services offered by the multicast protocol.

2.3.1.2 Sender or Receiver Oriented

In a sender-oriented approach, the sender is responsible for ensuring that all packets have been successfully received by the receivers i.e. the sender is responsible for data loss detection and recovery. In a receiver-oriented approach, the receiver makes sure that it has received all packets sent by the sender i.e. the receiver is responsible for data loss detection and recovery [3].

In a sender-oriented approach, the receiver sends an ACK for every packet sent by the sender. This leads to an ACK implosion and as result this approach does not scale well to a large number of receivers [3]. In addition, the sender would need to keep track of all the receivers in a multicast group. In a multicast environment, obtaining the list of receivers might be difficult [2]. In a receiver-oriented approach, the receiver sends a NACK for any missing packets from the sender. This leads to less network traffic. But receiver based protocols are inherently more difficult to implement. Some of the receiver-based protocols that exist are tree-based multicast protocol, which uses designated receivers to send missing data to its members [1], log-based multicast protocol which use primary and secondary log servers to send missing data instead of the data source, etc.

In addition to the above, there are some design considerations which are common to both the sender or receiver-oriented approach such as caching or buffering, retransmissions mechanisms, duplicate transmissions and scheduling of repair and data packets [2][3].

2.3.1.3 Congestion Control

Congestion control is an important design consideration for development of a multicast protocol. For example, should the sender adapt its data

transfer rate to the slowest member in the receiver group or should that receiver be pruned from the member group [2]. In the tree-based multicast protocol implemented as part of the JRMS, the protocol on detection of network congestions, drops its data transfer rate by a given percentage [1].

2.3.1.4 Network Topologies

The multicast protocol should be able to adapt to the different underlying network topologies.

2.3.1.5 Security, Authentication and Privacy

In addition to scalability and reliability, the multicast protocol can provide additional services such as security, authentication and privacy. In JRMS, these services are distinct from the multicast transport protocols and are provided as part of the multicast service framework.

2.3.2 Simple Reliable Multicast API

As part of this project a Simple Reliable Multicast protocol will be developed. Some of the considerations for the protocol are

- The protocol will be sender-oriented protocol in order to keep the implementation simple.
- The protocol should support a one-to-many or many-to-many application.
- Loss detection and recovery will be provided by a ACK based scheme. The receiver will send an ACK for every packet received from the sender.
- Application data will be buffered at the sender.

-
- An API library will be developed to decouple the protocol from the application.

2.4 Application

As part of the project an application will be developed to use the Simple Reliable Multicast API and the modified M2MP protocol to send and receive packets. The application will provide a visual display of the packets sent and received.

3.0 Project Analysis

As part of the project, the reliable multicast API will be tested in an unreliable M2MP network environment under varying degree of unreliability and throughput delay. The throughput of the application using the multicast API will be evaluated under varying degree of unreliability and throughput delay.

The DropPacket and Throughput-Control layer will be used to simulate an unreliable wireless network.

4.0 Project Architecture

The proposed architecture of the M2M protocol to simulate an unreliable network is as shown in Figure 1. The DropPacket layer will be added to the M2M protocol stack. This layer would be a M2MP Channel implementation. The Throughput-control layer will be part of the architecture [10]. The DropPacket layer will be added before the Throughput-control layer. Doing this, will allow the application to control the throughput for the data packets that will be broadcast on to the network. Also, the design and implementation of the Throughput layer is tightly integrated with the Channel layer in the M2M protocol stack [10].

In addition an application will be developed which will use the reliable multicast API to send packets over the M2MP network.

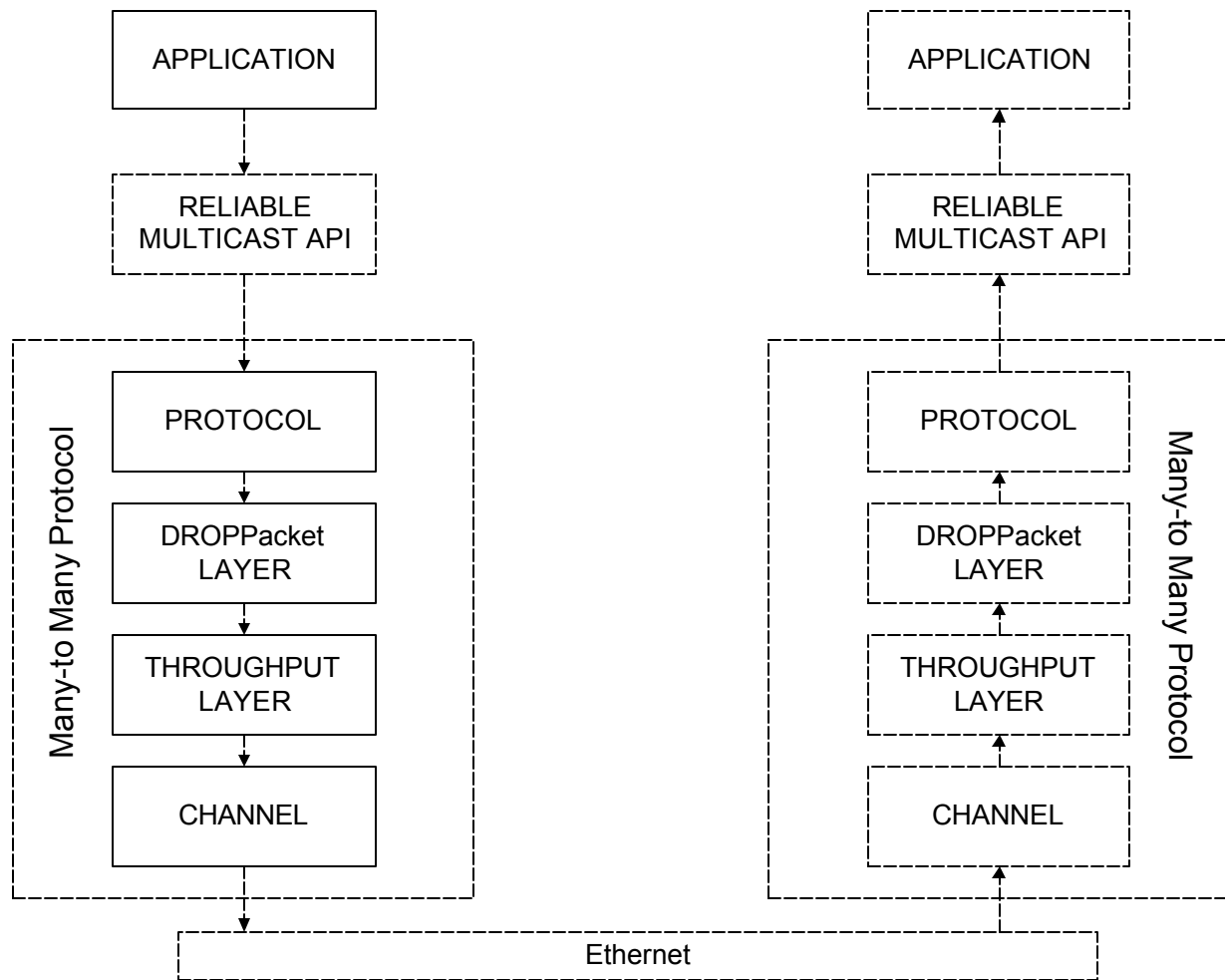


Figure 1: Proposed Project Architecture

Bibliography

- [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”, IEEE/ACM Transactions on Networking. Electronically available on the Internet at <http://citeseer.ist.psu.edu/495705.html>. Downloaded April 2003.
- [3] Gemmell, Jim., Leibeher, 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.experimentstuff.com/sunr/techrep/1998/abstract-68.html>. Downloaded March 10th, 2004.