1997

Applied reliable multicast using the Xpress Transport Protocol (XTP)

Johnstone, George S.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPLIED RELIABLE MULTICAST
USING THE XPRESS TRANSPORT PROTOCOL
(XTP)

by

George S. Johnstone
Glenn D. Williams

March, 1997

Thesis Advisor:	**W. Timothy Strayer**
Co Advisor:	**Rex Buddenberg**

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1997	Master's Thesis

4. TITLE AND SUBTITLE
APPLIED RELIABLE MULTICAST USING THE XPRESS TRANSPORT PROTOCOL (XTP)

5. FUNDING NUMBERS

6. AUTHOR(S)
Johnstone, George S., and Williams, Glenn D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Reliable multicast protocols provide a means to deliver data from one sender to many receivers with assurance. Reliable multicast is better suited than unicast for the bandwidth restricted, high error rate, hostile communications environment found in the military's tactical arena. General purpose protocols ensure adaptability to the variety of communications suites currently used by the military. As well, any acceptable multicast protocol must support varying levels of assurance, from unreliable delivery to full reliability.

This thesis evaluates the performance capabilities of one implementation of the Xpress Transport Protocol -- SandiaXTP, which is a reliable multicast transport protocol. Four experiments are run on a testbed consisting of four Sun SPARC4 workstations. These experiments look at unicast and multicast transmissions with varying numbers of induced errors. The included performance measurements examine the various challenges present in a communications medium subject to attack. The results demonstrate that reliable multicast in a tactical environment is possible.

14. SUBJECT TERMS
XTP, Multicast, Reliable Multicast, Telecommunications Networks, Network Management, Network Administration, Transport Protocols

15. NUMBER OF PAGES 135

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited

APPLIED RELIABLE MULTICAST USING THE XPRESS TRANSPORT PROTOCOL (XTP)

George S. Johnstone
Lieutenant, United States Navy
B.U.S., University of New Mexico, 1989

Glenn D. Williams
Lieutenant, United States Navy
B.A., University of Colorado, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 1997

Authors: [redacted]

George S. Johnstone

[redacted]

Glenn D. Williams

Approved by: [redacted]

W. Timothy Strayer, Thesis Advisor

[redacted]

Rex Buddenberg, Thesis Co-Advisor

[redacted]

Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

Reliable multicast protocols provide a means to deliver data from one sender to many receivers with assurance. Reliable multicast is better suited than unicast for the bandwidth restricted, high error rate, hostile communications environment found in the military's tactical arena. General purpose protocols ensure adaptability to the variety of communications suites currently used by the military. As well, any acceptable multicast protocol must support varying levels of assurance, from unreliable delivery to full reliability.

This thesis evaluates the performance capabilities of one implementation of the Xpress Transport Protocol — SandiaXTP, which is a reliable multicast transport protocol. Four experiments are run on a testbed consisting of four Sun SPARC4 workstations. These experiments look at unicast and multicast transmissions with varying numbers of induced errors. The included performance measurements examine the various challenges present in a communications medium subject to attack. The results demonstrate that reliable multicast in a tactical environment is possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, we acknowledge our families. Without your continued encouragement and support, we would not have achieved our past successes nor would we have reached this milestone in our lives.

We wish to express our sincere appreciation to Dr. W. Timothy Strayer. It is not often that one meets truly inspiring people in life. Your leadership, motivation and excitement were the driving force that maintained the research focus and nurtured this thesis to completion. You even made it…fun?

We acknowledge and thank Rex Buddenberg for planting the seed that became this project. Your ability to bring together teams and projects guides the progress of next generation protocols and will ultimately advance the way the military communicates.

We are grateful to Steve's wife, Kathy, for editing, proofing and being our biggest supporter. Thanks for putting up with us.

Steve thanks his mother, Rose Ford, and his brother, Warren Johnstone, for a lifetime of love and encouragement. Steve also thanks his best friend, partner and loving wife, Kathy for the joy she has brought to his life. Additionally, Steve sends a special thank you to Sandra Hollingsworth for keeping his family sane through all this.

Glenn thanks the best father anyone could ever hope to have, Charles H. Williams, whose lifetime of caring and consideration is unsurpassed. Glenn also expresses his sincere thanks to Linda Gilmore for her kind words and loving personality.

I. INTRODUCTION

A. INTRODUCTION

Data may be transferred from a host to multiple recipients by using either a multiple unicast or single multicast transmission. A unicast transfer of data may fulfill the needs of information exchange between receivers on an individual basis. However, in a military combat environment, there are numerous scenarios in which multiple receivers need data at the same time. It is under these operating conditions that a multicast protocol is most beneficial, facilitating the simultaneous dissemination of information to specific receivers within the network over a single connection. This thesis provides an analysis and evaluation of several existing US Naval communication methods, and examines the potential for adoption of reliable multicast within a tactical amphibious scenario.

This chapter presents the differences between multicast and unicast communication. To clarify the issue of multicast and unicast, a brief description of both is provided. This is followed by a description of the existing Navy communications technologies and descriptions of the various multicast protocols. A portrayal of the data transfer protocol stack follows, and then a discussion of the reliability issue. Finally, the advantages of multicast in a tactical environment is briefly described.

B. UNICAST DATA TRANSFER

A *unicast* packet is a packet addressed to a single node on the network. Each node has a unique address; packets are forwarded by routers until they arrive at the destination. An IP address contains an encoding of the subnet, which assists the routing protocols. It is only those routers that are in the transmission path of the packet that will forward the packet. (Figure 1.1).

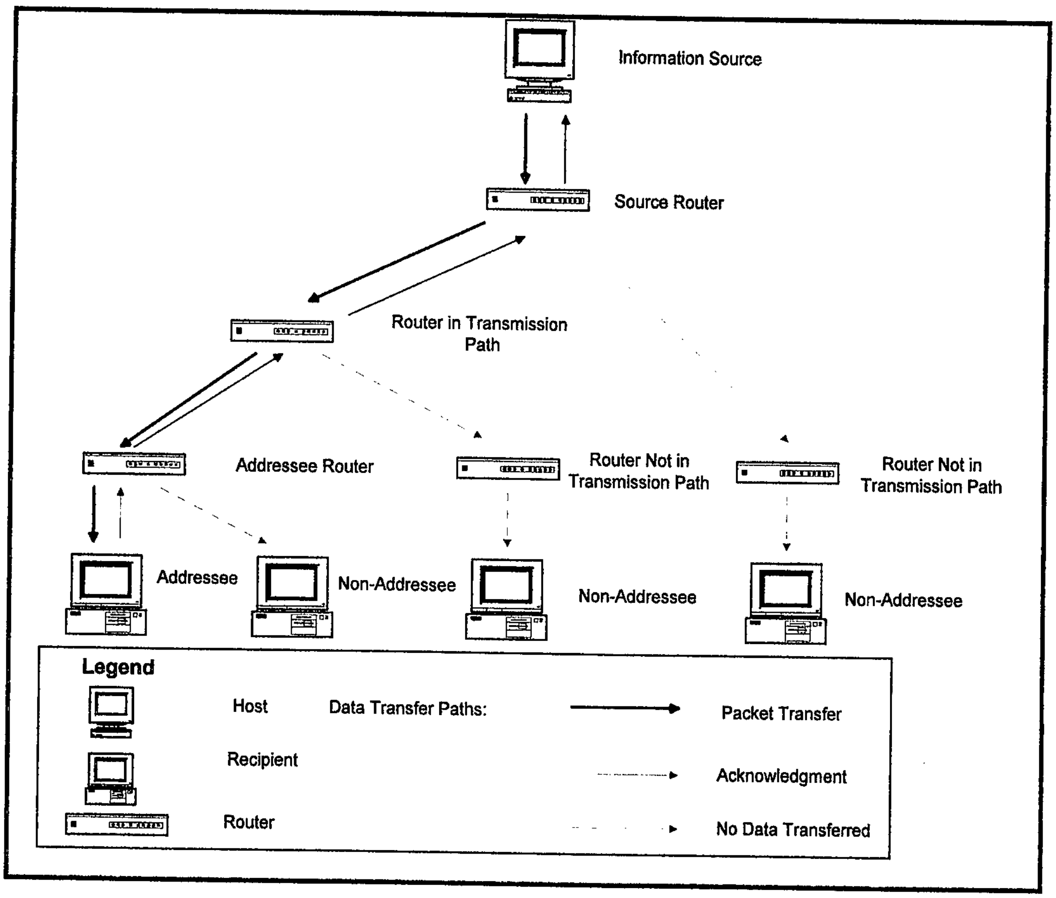

Figure 1.1 Example of Unicasting

C. THE ISSUE OF BROADCASTING DATA

A second method of data transfer is to broadcast the data. This entails each bridge or router forwarding the received packets on all interfaces with the exception of the path from which the packet was received. This is similar to a television broadcast, in that the signal will be retransmitted without regard to which receivers are interested in the data.

This results in the indiscriminate use of bandwidth, as ALL destinations will receive the packets which must traverse all data transmission routes, even if there are no destination addresses downstream (Figure 1.2).

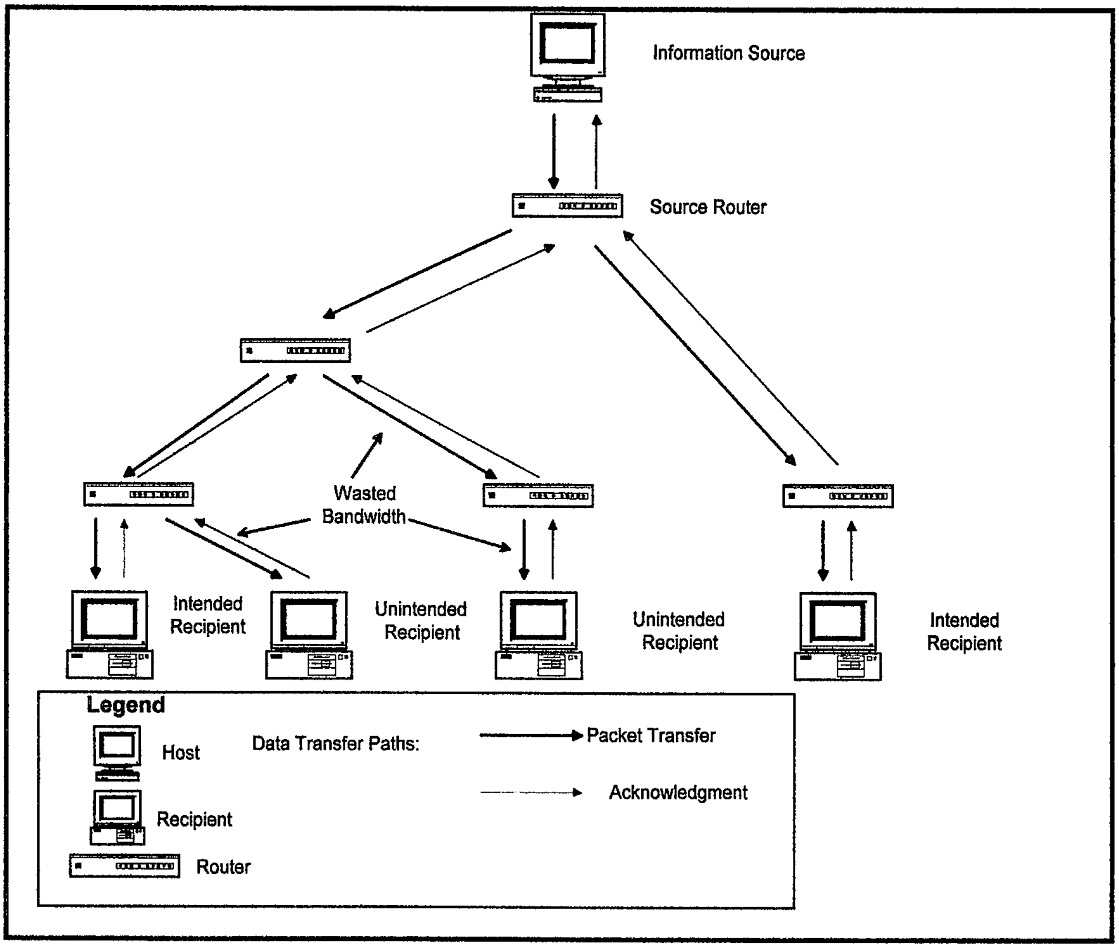

Figure 1.2 Example of Broadcasting

D. MULTICAST DATA TRANSFER

A *multicast* packet is a packet addressed to a subset of nodes on the network. The group of nodes which share the same multicast address comprises a multicast group. Bridges forward these multicast packets to the downstream interfaces (i.e., all segments "see" all packets). Data are delivered to each member of the multicast group. This transmission method necessitates that only a single copy of the information is sent by the source, and the routers within the transmission path generate the required number of copies for delivery to all members of the multicast group. Multicast possesses several benefits over unicast, including:

- A single transmission by the source is sent to numerous recipients;
- More efficient use of bandwidth as transmission over any given route only occurs once;
- Concurrent receipt of data;
- Enhanced mobility, since the destination multicast IP address is not tied to the subnet;
- Routing is discovered by Multicast Backbone (MBONE) protocols;
- Packets are only delivered to those interested.

Reliable multicast suffers from two disadvantages: throughput is determined by the slowest receiver, and group management becomes more difficult as the number of receivers increases. A visual representation of multicasting is provided in Figure 1.3.

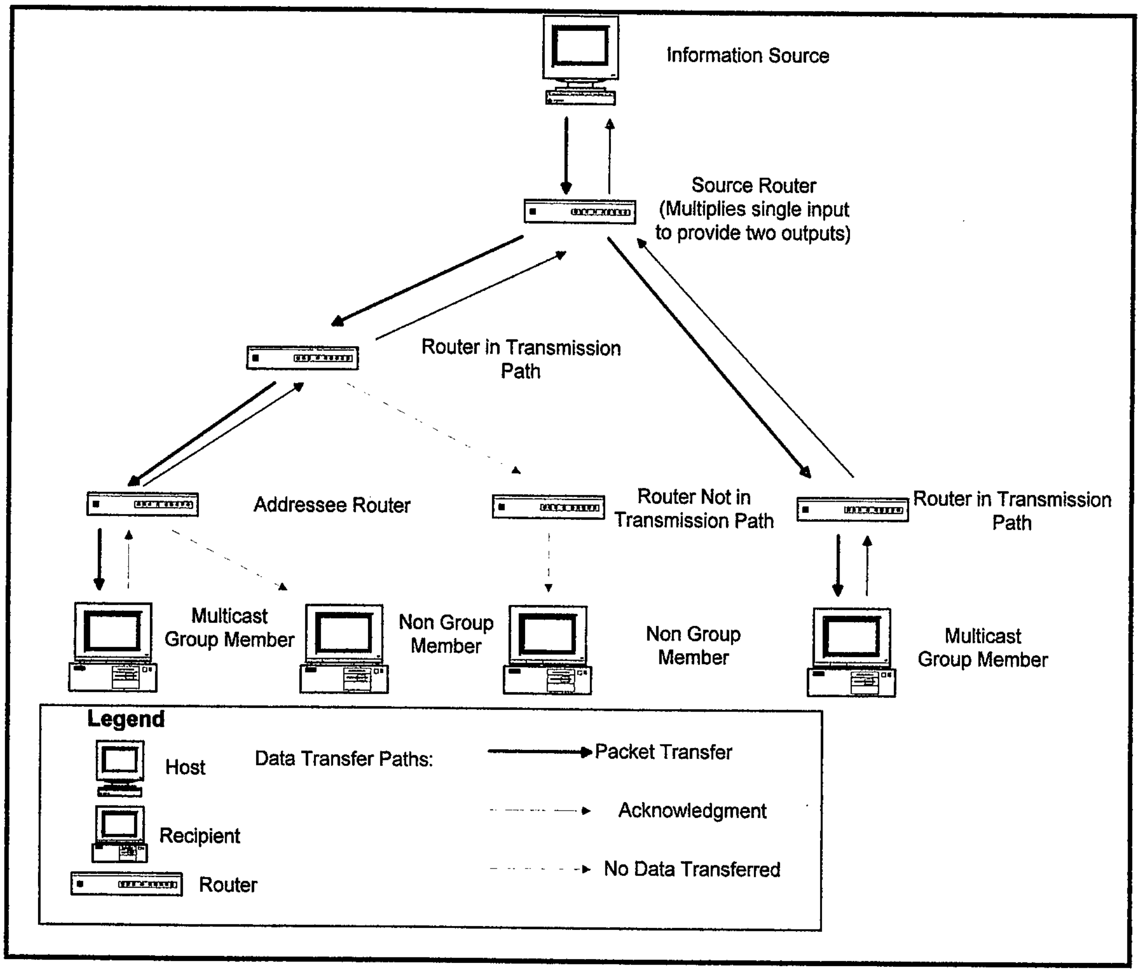

Figure 1.3 Example of Multicasting

E. EXISTING NAVY TECHNOLOGY

1. Ship To Shore Circuit Modes Of Operation

There are three methods of ship to shore radio frequency circuit operation: duplex, simplex, and semi-duplex. Hardware and frequency availability dictate which method will be used.

a. Duplex

A duplex circuit is designed to transmit and receive data simultaneously. The transmitting and receiving stations each utilize two frequencies: a station transmits on one frequency, and receives on the other. The advantage of this type of circuit is the time consideration: one frequency can be used for continuous data transfer while a second frequency can be utilized for packet acknowledgment.

b. Simplex

A simplex circuit consists of a single frequency on which data is both transmitted and received. The least technologically advanced of the modes of operation, simplex circuits are normally reserved for UHF and those platforms that are not equipped with duplex equipment.

c. Semi-duplex

A semi-duplex circuit is a combination of duplex and simplex modes of operation, and is used primarily on task force/task group/ORESTES. With the exception of the Net Control Station (NECOS), all stations utilize simplex procedures. The NECOS transmits and is received on a second frequency, allowing for continuous transmission by the NECOS.

2. Fleet Communications

The highly dynamic operating environments characteristic of modern fleet units mandate the utilization of communications systems capable of providing high-speed, accurate, and secure two way data transfer.

Through equipment design and installation, many equipments are compatible with each other and can be used to accomplish various functions. Using this design concept, nearly all the communications needs of a ship can be met with fewer pieces of communications equipment than were previously required. (NAVEDTRA 1994)

The following is a brief description of these Naval assets.

a. Ultra-High-Frequency (UHF) Systems

The UHF band occupies the 300 MHz to 3 GHz band of the RF spectrum and is used for line-of-sight, short range communications. This means that the transmitting and receiving antennas must be physically visible to one another, and unaffected by the curvature of the earth. Since satellite communications are also line-of sight, the UHF band is utilized for both satellite uplink and downlink communications, effectively eliminating the possibility of direction finding in a hostile communications environment.

b. Super-High-Frequency (SHF) Systems

The SHF band of 3 to 30 GHz is used exclusively for line-of-sight communications, and primarily for satellite communications. "SHF satellite communications is a high-volume system that offers reliable tactical and strategic communications services to U.S. Navy elements ashore and afloat." (NAVEDTRA 1994)

c. CUDIXS

"The Common User Digital Information Exchange System (CUDIXS) provides a bi-directional, ship-to-shore-to-ship, high-speed digital data communications link between a ship and a Naval Computer and Telecommunications Area Master Station (NCTAMS) or Navy Computers and Telecommunications Station (NAVCOMMTELSTA)." (NAVEDTRA 1994) A single Fleet Satellite Communications (FLTSATCOM) half-duplex channel provides a synchronous link between the CUDIXS shore station and the subscribers afloat. The system is limited to 60 subscribers: 10 special subscribers and 50 primary subscribers. Special subscribers have the capability to transmit and receive data to and from CUDIXS; primary subscribers are allowed a "send only" capability. "The CUDIXS/NAVMACS (Naval Modular Automated Communications System) combine to form a communications network that is used to transmit general service (GENSER) message traffic between ships and shore installations." (NAVEDTRA 1994) NAVMACS acts as the automated shipboard terminal for interfacing with shore based CUDIXS systems and the Fleet Broadcast System. While a satellite communication circuit is an effective means of broadcasting and even multicasting data, CUDIXS is inappropriate for communications within a tactical group due to the reliance on a NCTAMS or NAVCOMMSTA. Additionally, the relatively small number of stations given the capability to transmit and receive data – ten – limits the overall functionality of CUDIXS.

d. TADIXS (Tactical Data Information Exchange System)

> TADIXS is a direct communications link between command centers ashore and afloat. TADIXS provides one-way transmission of data link communications. (NAVEDTRA 1994)

Unfortunately, the one-way TADIXS link does not fulfill the need for a reliable means of data transfer. Additionally, the exclusion of any units other than command level mandates a hierarchical information transfer procedure, resulting in time delays in delivering vital information to the operating forces. These delays could be the critical moments that differentiate between success and failure in a military operation.

e. Fleet Broadcasts

Fleet broadcasts are the primary means by which mobile units receive messages. The method by which these transmissions are made has been termed the Fleet Multichannel Broadcast System (MULCAST). Also known as the "November System," the highly flexible MULCAST system provides global service to the fleet via one of the four major NAVCOMMAREAs. (NAVEDTRA 1994) The areas of coverage are shown in Figure 1.4, including their NCTAMS (in Guam, Honolulu, Norfolk, and Naples, Italy).

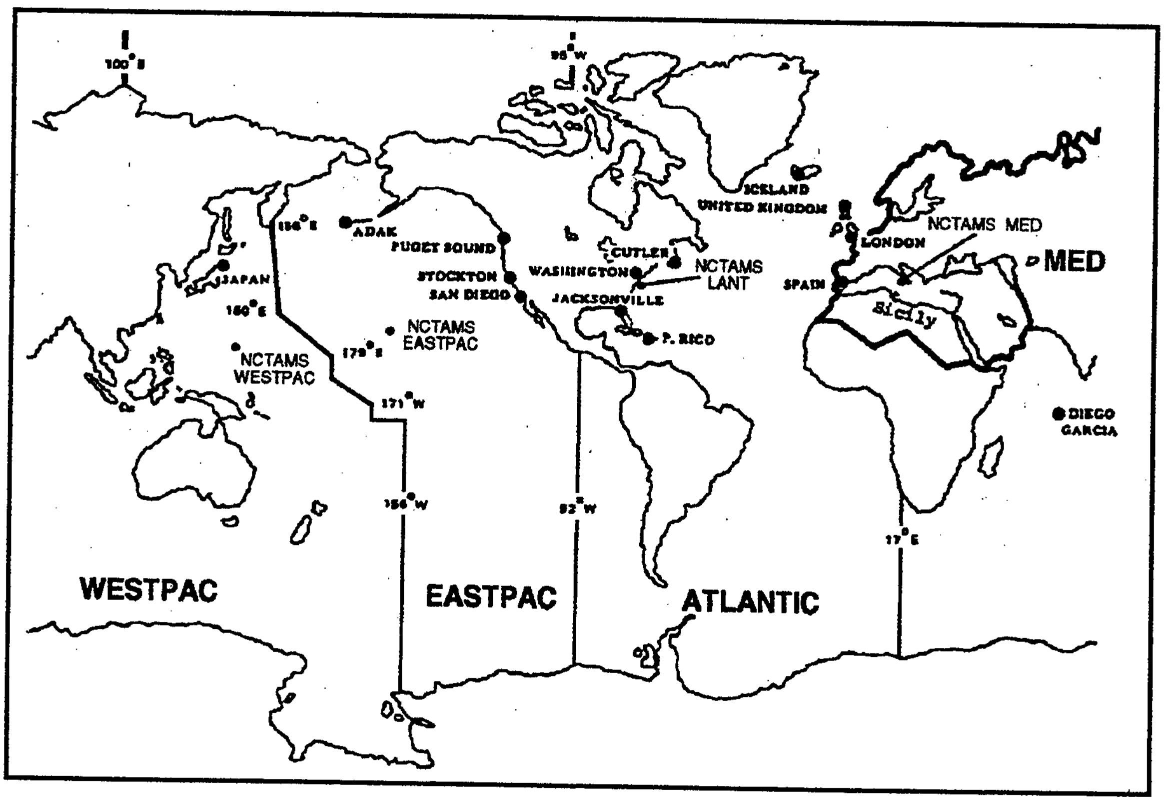

Figure 1.4 Naval Communications Areas (from NAVEDTRA Radioman Communications, 1994)

Each of the four NAVCOMMAREAs is controlled by a NCTAMS which supervises the coordination of the fleet broadcast and all other communication circuits assigned to that area. To receive MULCAST transmissions, which encompass 16 separate information channels, ships are assigned to copy specific channels, dependent upon ship type and similarity of mission. Further, ships are assigned to copy a channel for specifically addressed traffic and a channel for general traffic.

To ensure traffic continuity, each message broadcast via MULCAST is assigned an alphanumeric Broadcast Sequence Number (BCSN). To verify that all messages addressed to one's ship have been received, ships guarding the broadcast must maintain a file by BCSN listing all broadcast numbers transmitted. Additionally, once an hour on the hour, a message summary (RECAP) is transmitted which provides a summary of the traffic transmitted during the previous hour. The RECAP details the BCSN, precedence, date-time group (DTG), originator, and broadcast addressees for each message transmitted. If a ship did not receive a message for which it was addressed, it must request a retransmission of that message, or obtain a copy from a ship in company.

MULCAST is also inappropriate for tactical group communications as all transmissions must originate at a NCTAMS. Further, MULCAST is a non-reliable system: message receipt verification is accomplished via the RECAP, a serious detraction to the smooth flow of data, and a possible contributor to battlefield communications confusion.

f. NTDS

> The Naval Tactical Data System (NTDS) is a practical approach to ensuring the most effective use of both decision time and full fleet

capability. Integrated design and major components of the NTDS provide the fleet with automated data handling capability. (NAVEDTRA 1993)

In an operational environment, the computer-controlled NTDS coordinates the collection of data from the ship's sensors and from external sources via communications links. Inputs to the system include radar, sonar, navigation systems, electronic warfare, fire control, and manual inputs from keysets. As an output, NTDS provides a single display of air, surface, and subsurface contacts (and their threat evaluations), contact course and speed, weapons assignments, as well as land masses and the ability to determine bearing and range to any point within the display. There are four major subsystems within the NTDS system:

- Link 11: Link 11 provides high-speed computer-to-computer transfer of tactical environment information, command orders, and participating unit status to all other tactical data systems with a nominal range of 300 miles. Link 11 uses groups of participating units and a standard message format for the exchange of digital information among land-based, shipborne, and airborne platforms. Although primarily utilizing the HF and UHF bands, Link 11 may also be operated by Limited Range Intercept Satellite Communications (LRI SATCOM), but this increased advantage is offset by imposed restrictions, as system architecture mandates the use of a task force aircraft and limits Link data transmissions to one channel, along with two channels for voice communications. In addition, the Link 11 system is based on obsolete equipment that has been in use for over twenty years, and is not portable to current technology. However, due to its high level of functionality, Link 11 is the preferred mode of tactical data transfer between operating units.

The scene of operations from a radar point of view is expanded by infusing data received from other Link 11 ships onto one plot. The infusion is synchronized via computer based on the position (latitude and longitude) of one's own ship, and the position of the other ships linking data. Unfortunately, errors are introduced by the unavoidable degradation of operating programs. Additionally, the plot is usually placed onto a paper trace by manual means. This plot accumulates errors resulting from not only incorrect source data as described above, but also plotting gear inconsistencies and human error.

- Link 14: Link 14 provides a means of transmitting track information, identity, engagement status, drop track reports, and gridlock information to ships not capable of participating in the Link 11 net. This system allows for the transmission of data via voice and teletype communications to those units not equipped with the more expensive, combat operations related hardware that makes up the Link 11 network. The data received must then be plotted manually, with the inherent flaws of the paper trace as detailed in the description of Link 11.

- Link 4A:

> Link 4A enables an operational program to take control of the autopilot in a suitable equipped aircraft to control landing and takeoff, to pursue, or to follow collision intercept geometry. It can control a flight to the strike area and return it to the base without requiring pilot action. (NAVEDTRA 1993)

The level of control provided by the Link 4A system is at the discretion of the pilot, who can opt for fully automatic operation, use of the visual display (only), or totally disregard the system. This system is very useful in relaying geographic and contact data to aircraft.

- Link 16: Link 16 is basically the same as Link 11, with the inclusion of a merged message format which allows for interservice and NATO link operations. Yet Link 16 faces the same limitations as Link 11, and can be further hampered by US forces infusing data from foreign forces which periodically rely on sensors of inferior accuracy.

g. SATCOM Systems

"Satellite communication systems satisfy many military communications requirements with reliable, high-capacity, secure, and cost-effective telecommunications." (NAVEDTRA 1994) While allowing for worldwide coverage (with the exception of those areas above 70°) including underway platforms, satellite systems also provide an alternative to large, fixed land-based systems. There are two types of satellite communication systems: active and passive. An active satellite receives an uplinked signal, amplifies the signal, and then retransmits the signal back to earth (downlink). A passive satellite simply acts as a reflector (or "bent pipe"), physically redirecting the uplinked signal back to the surface of the earth. A basic satellite system (with various types of transceivers) is shown in Figure 1.5.

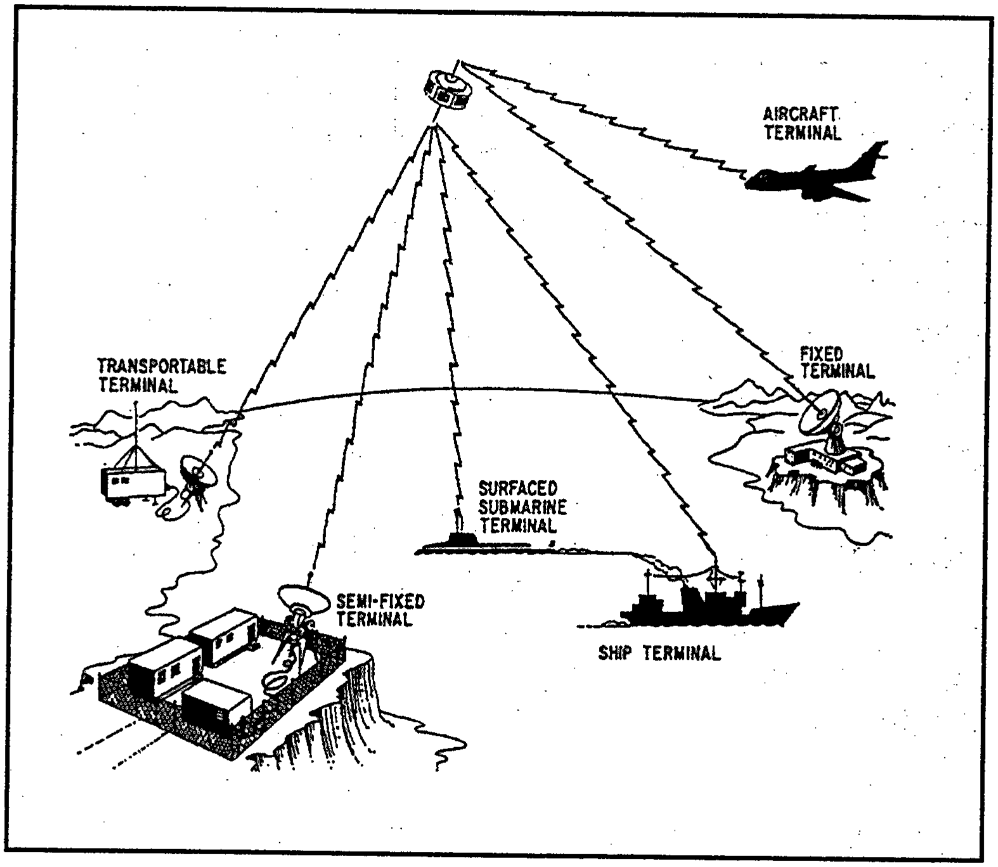

Figure 1.5 Satellite Communications Systems (from NAVEDTRA Radioman Communications, 1994)

A satellite communications system is ideal for a tactical assault scenario since the satellite is not susceptible to localized attack. With a transceiver operated by each of the members of a multicast group, communications delivery and receipt (at each of those locations) is primarily dependent upon the satisfactory operation of local routers and the transceivers themselves. The combination of a satellite communications system and reliable multicast provides greater overall communications efficiency while keeping transmission errors to a minimum. Therefore, multicast communications within the scenario presented in the following chapter are the central focus of this thesis, with an emphasis on the conduct of experiments which vary factors such as:

- Unicast or multicast transmission
- Number of multicast group members
- Inclusion of induced packet transfer errors

F. THESIS ORGANIZATION

This thesis is targeted for technical and non-technical readers. A short description of each chapter of this thesis is provided below as an overview:

1. Chapter Descriptions

- CHAPTER I – INTRODUCTION – Contains an overview of data communications technology, existing Navy communications technology, and the reasons for adoption of a new multicast system.
- CHAPTER II – PROBLEM STATEMENT – Describes the tactical scenario in which the multicast system is to be implemented. A proposed solution is presented, along with descriptions of existing multicast protocols and the decision to use Xpress Transport Protocol (XTP). Also included is a list of related work.
- CHAPTER III – EXPERIMENTATION ENVIRONMENT – A description of the simulation environment is presented, including the "ships to chips" paradigm, and the identification of those pieces of hardware to be utilized during the multicast transmission experiments.
- CHAPTER IV – EXPERIMENTS – A description of the various experiments to test the feasibility of amphibious scenario multicast. Experiments include unicast, multicast, increasing the number of multicast receivers, and the inclusion of packet errors. Detailed analysis of the experiments described in Chapter III. Both visual and technical analysis will be presented.

- CHAPTER V – CONCLUSIONS AND RECOMMENDATIONS – A summary of the importance of military multicast, and its potential for increasing tactical awareness. Finally, guidance on possible future expansion of the included research is presented.

2. How to Use this Thesis

Because there are both military and civilian entities involved in the acquisition of military hardware and software systems, recommendations on how to use this thesis for both types of employment are listed below:

a. Operational Forces

These individuals include assault team communications personnel, CIC personnel, ship commanding officers, amphibious group commanders, fleet commanders, and fleet CINCs. The key players in any combat operation, it is these individuals who evaluate the current systems and provide recommendations for improvements or replacements. This thesis demonstrates the need for a tactical scenario multicast system, and provides one answer to the question "What system should we adopt?"

b. Program Management

These individuals include members of the government who provide funding for systems research and acquisition, and the private contractors who develop the requested systems. For these persons, this thesis provides a feasibility study into the potential use of a multicast system for close-in amphibious operations. Additionally, analysis is provided of the potential benefits to be gained from the adoption of that system.

II. PROBLEM STATEMENT

A. INTRODUCTION

The US Navy has been using multicast transmission for audio and video applications. Until now, technology did not support the reliable transfer of multicast data, nor was this necessary: the loss of a few bits or packets in audio/visual data transfer yields a negligible impact for those applications. However, now the technology exists to support reliable multicast data transfer, and the Navy should incorporate its use in the tactical environment. Developments in the speed of CPU's, establishment of the IP Multicast Backbone (MBONE) on the Internet, and the rapid growth in the number of users of Internet-style communications systems have all contributed to the need to reliably transfer large amounts of data to multiple addresses simultaneously. Experimental protocols exist that possess the potential to solve the reliable multicast problem. These will be discussed later in this chapter.

There are several scenarios in which the military could benefit from multicast data transmission of battlefield images and text. It is the intent of this chapter to provide one such basic scenario: an amphibious beach landing and the subsequent periodic transfer of landing zone conditions. The communications that occur between the ships and the tactical commanders on those ships is not the direction of this study; instead, this scenario is intended to motivate reliable the multicast transfer of information between the Naval Beach Group Element (ashore) and the ships at sea. This scenario will serve as the basis for the multicast experiments described in Chapter IV. To support those experiments, an analysis of existing multicast protocols is included, and the choice of which protocol best supports the Navy's needs is tendered.

B. AMPHIBIOUS READY GROUP SCENARIO

Reliable multicast could be applied to many military scenarios, but this thesis focuses on one tactical application – the Amphibious Ready Group (ARG). The ship types and command structure that comprise an ARG may vary slightly, but the general composition remains the same.

Tactical orders are normally received by the Joint Task Force Commander onboard an aircraft carrier (CVN) which can be located hundreds of miles away. Orders are then passed to the Commander Amphibious Task Force (CATF) embarked aboard the senior vessel in the ARG, normally an Amphibious Assault Ship (LHD, LPD or LPH). Additionally, the ARG normally contains one Amphibious Transport Dock (LPD) and one or two Dock Landing Ships (LSD). Finally, an Assault Craft Unit consisting of three Landing Craft, Air Cushion vehicles (LCAC) will be carried by each LSD. These LCAC are used for high speed transfer of troops, equipment and cargo from the ships to designated areas ashore. There are other elements that round out the entire ARG composition; however, it is the new technology and capability introduced by the LCAC that brings forth one need for reliable multicasting within the Navy.

The Chain of Command involves the ARG Commander (CATF), Commanding Officers of the ARG ships, and the following embarked units stationed aboard the Amphibious Assault Ship:

- Commander of the Marine Expeditionary Unit (MEU), responsible for further dissemination of orders to the Marine Corps units on the LPDs and the LSD. This is the same person that assumes the role of CLF during the actual assault.

- The Tactical Air Commander.

- Commanding Officers of embarked squadrons and other Marine Corps and Navy elements that might be temporarily stationed aboard the assault ship.

Aboard the LSD, orders are passed to the Officer in Charge of the LCAC detachment. Aboard the LPD, orders are passed to the Air Detachment and the Naval Beach Group – the team of naval personnel that establish a foothold on the beach and transmit back to the ARG the beach, surf and weather conditions. A summary of the Chain of Command and the vessels involved is shown in Figure 2.1.

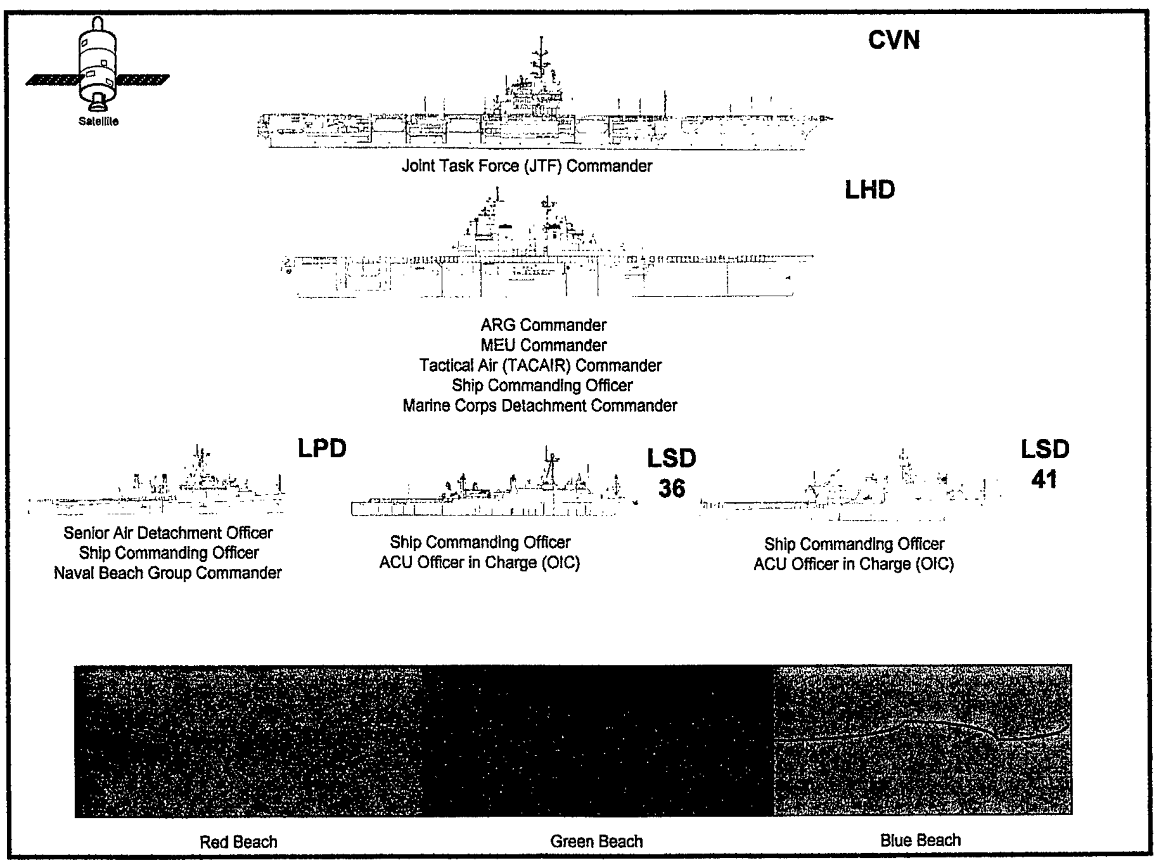

Figure 2.1 Amphibious Assault Scenario

C. PROTOCOL REQUIREMENTS FOR THE SCENARIO

There are many features in next generation transport protocols that are useful for communications between fleet units. However, it is important to narrow the field to a critical few that would be realistic to accomplish. As a result, this thesis will focus on several key attributes that are considered to be the basic features for a successful migration to a military environment.

First, it is imperative that any adopted protocol remain focused at the transport layer and not the application layer. Remaining focused at the transport layer means that the protocol can remain flexible and be easily adapted to varying communications requirements.

Second is the consideration of "total ordering" or "source ordering" of packets. Total ordering facilitates a causal relationship between the messages in the multicast group (i.e., if one message causes another message to be sent, the two messages are seen by a third entity in that order, and not the opposite order). This assumes multiple transmitters. (Strayer, 5 March 97 email to Johnstone) Even though total ordering has application in military communications, normally only one transmitter will act as the central point for multicast transmission generation. In this set of circumstances, source ordering is preferred, in which the sole transmitter places outgoing messages in the proper order. The class of applications presented by the scenario will only require source ordering.

Lastly, not all information that is sent in a tactical environment has a need to be delivered reliably; therefore the protocol must be able to support various transmission options (e.g., unicast, unreliable multicast and reliable multicast). These attributes will serve as a basis for protocol selection for forthcoming experiments.

D. THE OSI MODEL

In order to make an informed decision of which multicast protocol best suits the requirements of the scenario, it is necessary to understand the Open Systems Interconnection (OSI) reference model. In response to the impending inevitable proliferation of proprietary systems and their incompatibility with each other, the International Organization for Standardization (ISO) established a subcommittee in 1977 to develop an architecture that would define the communication tasks that define the ability of computers to exchange information. The result was the OSI reference model, adopted in 1983, which is a framework for defining standards for linking heterogeneous computers. The model is shown in Figure 2.2.

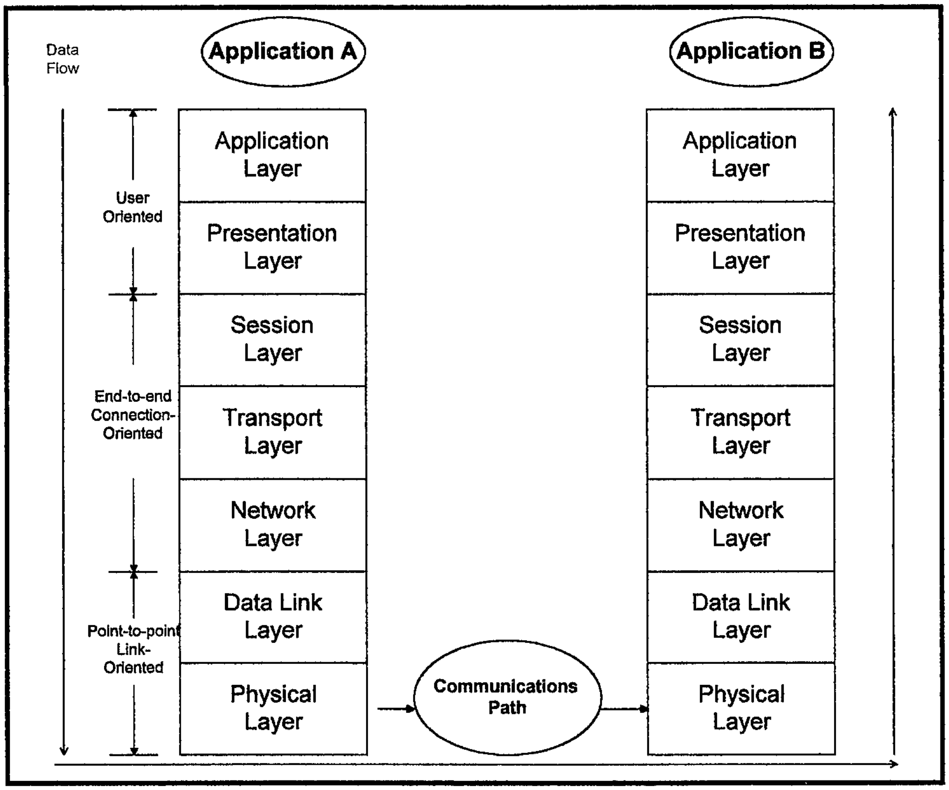

Figure 2.2 The OSI reference model

1. OSI Model Layers

The *physical layer* is concerned with the transmission of an unstructured bit stream over a physical communications medium. It deals with the mechanical, electrical, functional and procedural characteristics to access that medium. This layer covers the physical interface between devices and the rules by which bits are passed from one device to another. The *data link layer* "provides for the reliable transfer of information across the physical link; it sends blocks of data (frames) with the necessary synchronization, error control and flow control." (Stallings, 1994) Since the physical layer provides a means of transferring bits (only), the data link layer is needed to frame the bits into packets, provide some error detection, provide the means to activate, maintain, and deactivate the link.

The end-to-end service is provided by the network, transport, and session layers. The *network layer* "provides upper layers with independence from data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections." (Stallings, 1994) It provides the routing and relaying of network data units across multiple network segments and multiple networks. The *transport layer* "provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control." (Stallings, 1994) This layer provides a reliable mechanism for the exchange of data between processes in different end-systems by ensuring that the data units are delivered error-free, in the proper sequence, with no losses or duplications. The purpose of the *session layer* is to provide the means for organizing, synchronizing, and managing dialogues between end-nodes. The key services provided by the session layer include dialogue discipline (full duplex, half-duplex, etc.), groupings of data, and the ability to recover data if there is a failure between predetermined checkpoints within the data stream.

The user-oriented portion of the model consists of the presentation layer and the application layer. The *presentation layer* "provides independence to the application processes from differences in data representation (syntax)." (Stallings, 1994) The *application layer* "provides access to the OSI environment for users and also provides distributed information services." (Stallings, 1994) This layer contains management functions and other mechanisms to support distributed applications, such as file transfer and electronic mail.

The focus of this thesis is to demonstrate the benefits of adopting a reliable multicast protocol for data transmission. Doing so mandates a closer look at the transport layer, where a multicast protocol undertakes its mission.

2. The Transport Layer: A Closer Look

As discussed above, the transport layer provides for a reliable exchange of data between processes by ensuring that the data is transferred without errors, properly ordered, and without duplication. The transport layer may also be concerned with optimizing the use of network services and providing a requested quality of service to session entities, such as acceptable error rates, maximum timing delay, packet priority, and message security. Essentially, the transport layer serves as the user's liaison with the communications facility. (Stallings, 1994) "In many ways, the scenario that places the greatest responsibility on the transport layer protocol is that of providing end-to-end reliable data delivery over an unreliable underlying network layer service." (Strayer, Dempsey, Weaver, 1992)

3. Reliable Multicast in the OSI Model

The OSI model can be summarized and related to this thesis by sectioning the model into four parts, as shown in Figure 2.3. The highest section of the model, Applications, adds no value to the integrity or reliability of the bit stream, and exists solely to enable the transfer of data via requests to the XTP daemon. The next highest section of the model consists of a reliable multicast protocol, many of which are discussed in the following section. This protocol will be required to perform end-to-end error recovery and flow control. The next section consists of the Network Layer and for purposes of this thesis, will be the IP Multicast Backbone (MBONE) which was created to allow for the multicast transfer of data across the Internet. Finally, the Media layer is the collection of physical and datalink protocols (often packaged together, like Ethernet and FDDI) that provide the network infrastructure to deliver the multicast packets to the intended receivers.

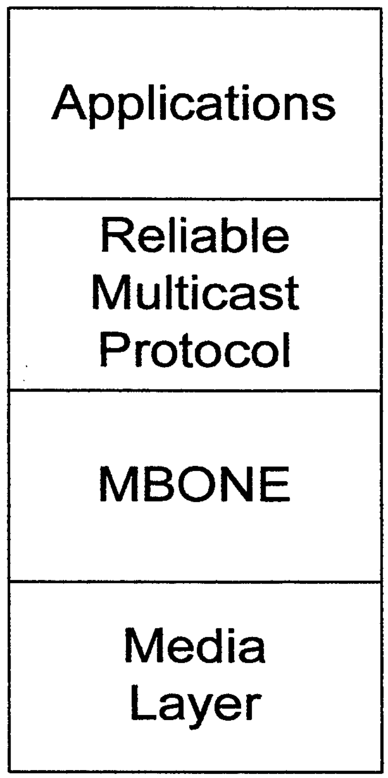

Figure 2.3 Abbreviated OSI Model

With an understanding of how the OSI Model functions to facilitate the transfer of data from one platform to other platforms, the question is now raised of which protocol should be used to fill the "Reliable Multicast Protocol" block in Figure 2.3. The following section provides a brief description of the most advanced multicast protocols, and briefly describes the advantages and disadvantages of each. This information is then used to make a decision of which protocol best suits the needs of the scenario.

E. RELIABLE MULTICAST PROTOCOLS

1. Definition of Reliability

As a precursor to discussing reliable multicast protocols, the term "reliable" needs clarification. Unlike the dictionary definition: "That can be relied upon: DEPENDABLE" (Webster's, 1996), the most comprehensive definition of reliability within the multicasting environment refers to three properties of a protocol, as described by Garcia-Molina and Spauster (1991). The first property is the length of time required to deliver a multicast message. The second property is termed "atomicity" and refers to all group members being required to deliver a reply to the initiating application within a specified time interval. The final property is ordering and covers the range from no requirement for specific priority addressing to members of the multicast group, to full delivery guarantees. These guarantees may persist regardless of remote host failures or operations within a hostile communications environment.

The most advanced multicast protocols in use today that should be considered are the Reliable Multicast Protocol (RMP), the Reliable Multicast Transport Protocol (RMTP), the Reliable Adaptive Multicast Protocol (RAMP), the Multicast Transport Protocol (MTP-2), and the Xpress Transport Protocol (XTP) (Bradner and Mankin, 1996). Each of these transport protocols are considered to be mature and have been implemented to certain levels on various platforms (Petitt, 1996). However, there are marked differences in how each accomplishes the task of assured multicast delivery of data.

2. Reliable Multicast Protocol (RMP)

RMP is based on the "Post-Ordering Rotating Token" model in which a token is rotated among group members and messages are ordered by the token site after they have been sent. (Montgomery, 1994) A token list is maintained by all members of the group and is updated upon the arrival or departure of group members. Advantages of RMP include:

- Ability to vary Quality of Service from unreliable to totally resilient on a per packet basis;
- Hosts limited to unicast capability may participate;
- The rotating token distributes processing load among group members;
- Ability to reconstruct the token ring following a network failure.

However, RMP is hampered by its high overhead, as both messages and acknowledgments are multicast, as are NACKs and repairs (Petitt, 1996). Packets associated with administrative functions such as joining and leaving the group are also multicast, further restricting the scalability of RMP. Finally, RMP is mainly meant for high-availability systems under a somewhat controlled internetwork, such as metropolitan area networks (MANs) and corporate intranets (Montgomery, 21Aug 96 email to Petitt). The flow control mechanism utilized by RMP severely degrades performance of the protocol under conditions of moderate packet loss, and is the primary reason why the protocol is restricted to low loss environments. Some experience with RMP has shown that its performance suffers over a WAN (Petitt, 1996).

3. Reliable Multicast Transport Protocol (RMTP)

"RMTP provides sequenced, lossless delivery of bulk data from one sender to a group of receivers." (Lin, 1996) The objective of RMTP is to "...guarantee *complete reliability* at the expense of delay" for applications hosted on a wide area network (Lin, 1996). The multicast delivery tree of RMTP is rooted at a sender; subtrees begin at special receivers called Designated Receivers (DRs) or Acknowledgment Processors (APs). These DRs are responsible for reliably delivering data to their subtrees (Petitt, 1996).

RMTP is well suited for the reliable delivery of bulk data in a non-delay important environment. RMTP differs from RMP in that RMTP is a single transmitter system, it localizes ACKs and NACKs, and it sends repairs either unicast or multicast, thus saving bandwidth.

4. Reliable Adaptive Multicast Protocol (RAMP)

RAMP was designed for collaborative-interactive applications hosted on "...all-optical, circuit-switched, gigabit..." networks. (Koifman, 1996) However, "...RAMP's design is also relevant for the next generation of packet switched networks." (Koifman, 1996) There are two modes of operation for the RAMP protocol: Burst Mode and Idle Mode.

a. Burst Mode

A burst of data is a series of packets which follow each other within some specified interval, this is true for both the Burst Mode and the Idle Mode. When the time between two consecutive packets exceeds this interval, the first packet marks the end of a burst while the second packet marks the beginning of the next burst (Koifman, 1996).

Burst Mode is intended to be used during times of high data transfer. The beginning of each burst is marked by a data packet with an ACK flag set. Upon completion of the burst, a single Idle message is sent to signify that the sender resides in a silent status until the beginning of the next burst. Receivers that have chosen to guarantee reliability must respond to the sender's ACK flag by returning the sequence number of the data packet which has the ACK flag set (Petitt, 1996). If the sender does not receive an ACK from the receiver, it retransmits the data. If no response is received after several retransmissions, the sender closes its control channel to the failed receiver (Koifman, 1996).

b. Idle Mode

Idle Mode is intended to be used in times of little data transfer. In this mode, there is no ACK flag set in the first data packet of a burst, nor is there a single Idle message transmitted after the last data packet in the burst. Instead, a series of Idle messages is multicast between bursts, each within a fixed time interval. If the time constraint of this interval is exceeded and neither a data packet nor Idle message is received, the receiver(s) unicast a Respond message to the sender. If the sender fails to respond to this message, the channel between the receiver and the (non-functional) sender is closed. In a similar manner, the continued connectivity to receivers is verified by periodic transmission of Idle messages by the receivers. If an Idle message from a receiver is lost, the sender closes the channel to that receiver. (Koifman, 1996)

The receiver is responsible for detecting and acting on failed connections in Idle Mode. Although the periodic Idle messages consume more bandwidth than the protocol overhead of the Burst Mode, they relieve the sender of the burden of processing acknowledgments (Koifman, 1996).

5. Multicast Transport Protocol (MTP-2)

MTP-2 is based on the concept of one multicast "master" and many producers and consumers. The master controls all aspects of group communications (Braudes, 1993). Packet transmission in MTP-2 is contingent upon possession of a token. To obtain a token, a producer transmits a unicast request to the master. The approved request spawns a serialized approval response (message number). A producer in possession of a message number then marks all packets of the message for which the approval was granted and multicasts the packets. The token is implicitly returned to the master with the final packet of the message.

There are many advantages of MTP-2, the first of which is minimal overhead. For each message sent, the protocol adds a unicast token request and confirm packet. However, for one-to-many multicasting, the master can be migrated to the producer to eliminate this overhead. (Petitt, 1996) Also, migration of the master from one machine to another is relatively simple: any suspected loss of the master is verified by receivers sending it a special packet. If the master does not respond, the members assume that it has failed and elect a new master. The new master then "...accumulates information about the status of all active messages ... from ... all responding ... members." (Bormann, 1994) Additionally, MTP-2 defines a dedicated unicast channel for each producer-consumer pair. Lastly, prioritization can be enacted for both token requests and messages.

MTP-2 relies on a receiver-initiated/sender-oriented error recovery method that severely restricts scalability. Although unicast NACKs consume less bandwidth than multicast NACKs, retransmission requests for missing packets may be made by multiple receivers, leading to multiple retransmissions of the same packet(s). Another concern within the military application is the limit of the number of messages that can be pending: 12. (Bormann, 1994) Lastly, MTP-2 cannot guarantee full reliability. The producers of data transmit the packets and then, after a retention period has elapsed, discard the data. If a NACK is received after this time has passed, no retransmission is possible. (Petitt, 1996)

6. The Xpress Transport Protocol (XTP)

"The Xpress Transport Protocol is a high-performance transport protocol designed to meet the needs of distributed, real-time, and multimedia systems in both unicast and multicast environments." (Atwood, 1996) Moreover, it is designed to provide a wide range of communication services built on the concept that orthogonal protocol mechanisms can be combined to produce appropriate paradigms within the same basic framework (Strayer, 1996). An orthogonal approach allows specific functions to be operated independently of other functions (e.g., error control activation and deactivation does not have an effect on other functions, such as flow control). Additionally, XTP is "transmitter driven": receivers that are a part of the group are controlled by the transmitter and respond as directed. Therefore, "XTP appears to be the most mature of the protocols presented for consideration" (Petitt, 1996). A closer look at the XTP specification reveals additional benefits, described below.

a. General Approach

The major advantage that XTP can offer the Navy is that of a general purpose protocol with the potential of providing all the communication protocol needs, such as reliable datagrams, unreliable datagrams and reliable multicast connections, required in a tactical environment. Another important consideration is that the XTP specification is not restrictive. There is no specific requirement for data to have one particular structure (XTP Forum, 1995). This allows for adaptability to various communications needs found in a complex communications theater. Additionally, XTP is efficient in its error handling and flow control, thereby supporting a dynamic infrastructure. XTP takes a general purpose approach that departs from the trend of most multicast protocols today. The majority of these protocols focus on application layer framing, while XTP remains focused at the transport layer to provide general services to all applications (Buddenberg, 1997). This general purpose approach is very attractive because it provides greater flexibility and support for different levels of group reliability and may specify that only certain receivers within a group be fully reliable (Petitt, 1996).

b. Group Management

XTP employs a group management technique which is an aggregation of the information from each of the receivers that belong to the multicast group. This capability is important in order to manage the membership status of the group. This management includes the creation and deletion of group members, monitoring of members joining and leaving the group, and whether or not a member is actively responding (Strayer, Dempsey, Weaver, 1992). This group management also allows for "late joining" which allows group members to join and leave an established group during actual data transmission without interrupting the flow of that data. The late member will receive data commencing at the time of the join, and will not be sent previously transmitted data.

F. SUMMARY

With the rapid expansion in the use of distributed computing services both ashore and afloat, the Navy must continually seek new and better ways to employ emerging technologies, thus ensuring all assets are fully utilized. The use of multicast, and more specifically, the use of reliable multicast in the tactical environment can help during high traffic periods typical of crisis situations. Reliable multicast can be utilized to transmit any information where assured delivery is critical. The Navy needs to pursue a general purpose multicast protocol that is flexible enough to handle both traffic that requires reliability and traffic that does not require reliability. In attempting to determine which available reliable multicast protocol the Navy should adopt as a standard, it appears that the Xpress Transport Protocol possesses the attributes necessary to fulfill all of the demanding needs of the Navy. Specifically:

- Transportation layer focus
- Source ordering

- Ability for unicast, multicast, reliable, and unreliable transmission

It is for this reason the authors have chosen to use XTP as the protocol for evaluating reliable multicast performance over unicast performance.

III. EXPERIMENTATION ENVIRONMENT

A. INTRODUCTION

A testbed was created to analyze the performance capabilities of XTP 4.0. The testbed supports current experimentation objectives while allowing for continued expansion and scalability as more in-depth testing of reliable multicast protocols becomes possible. The Sandia National Laboratories implementation of XTP 4.0 was chosen based on the implementation's ease of use, built-in performance tracking mechanisms, immediate availability, and easy accessibility to support information.

B. TESTBED HARDWARE

The testbed was built using existing hardware available in one of Naval Postgraduate School's laboratories. In particular, four identical Sun SPARC4 workstations running Solaris 4.0 were used, attached via ethernet to the main campus network that provided access to the Internet, as depicted in Figure 3.1. These four machines each represent a main receiver or transmitter from each of the ARG ships presented in the scenario from Chapter II. The Sun workstations were chosen primarily because they have built in support for multicast IP. Equally important was the ability to demonstrate that reliable multicast could be implemented on standard machines without significant upgrade. This is critical in the military arena where upgrades and equipment acquisition can sometimes be difficult. The selected software package also supports several PC-based operating systems. Testing on those platforms was beyond the scope of this thesis, but provides opportunity for follow-on research and project expansion.

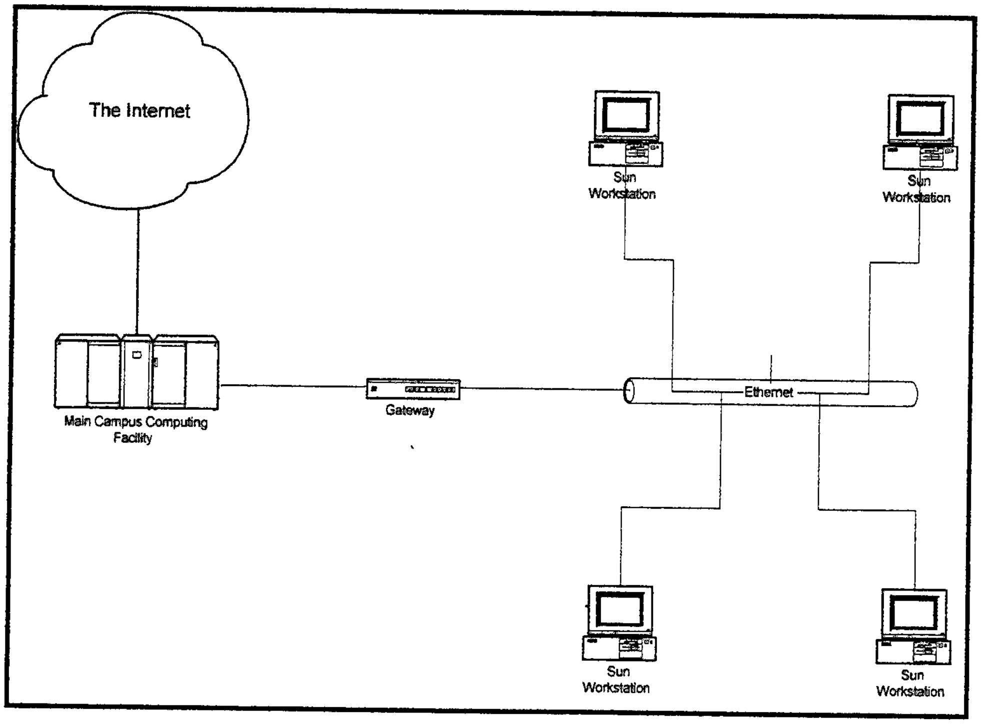

Figure 3.1 Experiment Testbed

C. SOFTWARE IMPLEMENTATION

1. Rationale for Selection of SandiaXTP

SandiaXTP, the Sandia National Laboratories implementation of XTP 4.0 (SandiaXTP website, 1995) was designed to provide a vehicle for research into the further development and advancement of the XTP 4.0 specification. As a result, SandiaXTP contains the latest changes to the specification agreed upon by the XTP Forum. Consequently, updates to the SandiaXTP implementation were frequent, and the test platform was continually updated with the latest available releases. The biggest gain realized by the SandiaXTP implementation of XTP 4.0 is its flexibility and platform adaptability. For a list of currently supported platforms, see Figure 3.2.

- ⇒ SGI workstations running IRIX 4.x and 5.x
- ⇒ Sun workstations running SunOS 4.x and Sun0S 5.x (Solaris)
- ⇒ DECstation 5000 running Ultrix 4.x
- ⇒ DEC Alpha workstations running OSF/1
- ⇒ HP workstations running HP-UX 9.x (but not with CC compiler)
- ⇒ I386 PCs running BSDI/OS 2.x
- ⇒ I386 PCs running FreeBSD 2.x
- ⇒ IBM RS/6000 workstations running AIX (but not with x1C compiler)

Figure 3.2 SandiaXTP Platform Compatibility List
From SandiaXTP User's Guide

2. Meta-Transport Library

The Meta-Transport Library (MTL) (MTL website, 1995) is a set of C++ base classes that serve as a basis for building transport protocols. SandiaXTP is built on MTL. While maintaining efficiency, MTL has been designed to be portable, adaptable, configurable and readable. The goal of MTL is to allow the rapid deployment and prototyping of a protocol implementation without the need for kernel modifications. Protocols are created from MTL simply by extending the base classes with specific protocol algorithms. Protocols based on MTL are implemented as user-space daemons (see Figure 3.3), which means code maintenance, debugging and customization are all easier since these activities do not require root access of kernel reconfiguration. Additionally, by being in the user-space, the need for root access to use the daemon is limited to changes in the daemon status (e.g., starting, stopping and checking the status of the daemon). This increases daemon availability to a greater range of users and sets the stage for portability to platforms running non-UNIX operating systems. (MTL User's Guide, 1997)

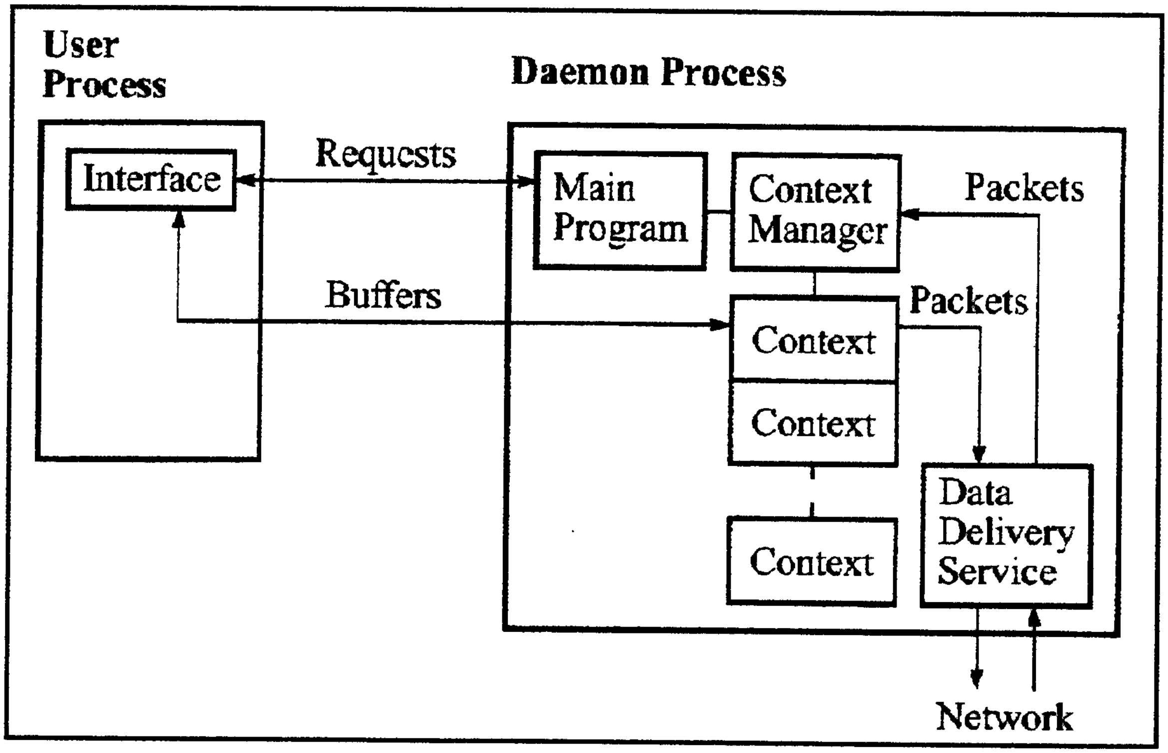

Figure 3.3 MTL User/Daemon Model
From Meta-Transport Library User's Guide

3. SandiaXTP

SandiaXTP is an MTL-based implementation of XTP 4.0. Since SandiaXTP is derived from MTL, SandiaXTP is a user space daemon where a user's applications make requests of the daemon, and the daemon satisfies them.

SandiaXTP exposes to the user the built-in protocol options that allow for the applications to create individual paradigms, such as unreliable datagrams and reliable datagrams, with error control, flow control and rate control based on the communication need. This means that the degree to which reliability is maintained rests solely with the user. This feature is particularly important in military applications where not all information would have the need to be transferred via reliable means. (SandiaXTP User's Guide, 1997)

4. Software Installation

The installation of MTL and SandiaXTP are both easy and straight forward. The user guide has detailed information to guide the user through the installation process. Since SandiaXTP relies on MTL base classes for its daemon services, MTL must be installed prior to SandiaXTP. Both software packages are available via anonymous FTP from "ftp://dancer.ca.sandia.gov" The user guide recommends that either the AT&T cfront compiler (CC) or the GNU C++ compiler be used. The latter is available via FTP from "ftp://prep.ai.mit.edu/pub/gnu." Both were used to install the software packages on different machines. Even though installation was just as easy using GNU C++, installing the compiler itself is not so easy. The AT&T cfront compiler is much easier to use if available.

The only major decision that has to be made when installing MTL is how to configure the software package. These options are clearly laid out in the user guide, and particular settings will be discussed further in the next chapter.

5. Software Settings

SandiaXTP was written to allow the user complete control over every aspect of the protocol to maximize performance. This control includes: if the machine will act as the transmitter or a receiver; setting the group IP address; sending a status request in the first packet; blocking on acknowledgment; setting fast acknowledgment; setting packet size; and setting buffer size.

a. Machine Function

There are four possibilities for this variable, all of which were used during the experiments detailed in Chapter IV. Setting the "-t" flag activates the machine as a unicast transmitter, while the "-r" flag activates the machine as a unicast receiver. Setting the "-T" flag activates the machine as a multicast transmitter, and the "-R" flag activates the machine as a multicast receiver. The receiver(s) were started prior to starting the transmitter, as at least one receiver must be available prior to activation of a transmitter.

b. IP Addressing

In unicast mode, an address must be specified, either an IP address or a DNS (Domain Name Server) address. In multicast mode, a Class D IP address must be specified. For our experiments, the default address (239.0.0.239) was used for all addressing.

c. Status Request (SREQ)

An SREQ is a bit that is set in a packet that requires the receiver of the packet to send back a control packet. The "-f" flag on the transmitter sends an SREQ in the "FIRST" packet sent by the transmitter. This flag causes the receiver(s) to acknowledge receipt of the initial packet, which tells the transmitter that the receiver(s) is/are active, thereby establishing the multicast group.

d. Block on Acknowledgment

The "-g" flag prevents the transmitter from sending additional packets (i.e., "block") until previously sent packets have been acknowledged by all receivers via an SREQ.

e. Error Control

To maximize throughput, these experiments were conducted with the "FASTNAK" flag ("-F") set. Use of the FASTNAK switch causes each receiver to send a control packet to the transmitter when an error is detected. Ideally, this control packet would reach the transmitter prior to any additional packets being transmitted. However, the high data transfer rates demonstrated by the protocol mean that packets sequenced for transmission after the packet in error may have been sent, resulting in GO BACK N retransmissions of the error packet *and all later packets*. Thus, GO BACK N describes a scheme in which once a control packet is received by the transmitter, that transmitter must *GO BACK* (in the transmission sequence) to the packet in error, and retransmit it, along with all successive packets.

D. SUMMARY

SandiaXTP's object-oriented implementation of the XTP 4.0 specification is designed to be a research protocol to further advance the development of the Xpress Transport Protocol. Although not intended for commercial use, its tracking mechanisms, immediate availability and local support base made it an excellent candidate for this research project. The protocol's reliability and adaptability were explored via a testbed that was built to support experimentation. Software installation is simple and supports a wide range of current computing platforms. Being a user-space daemon means that the protocol is highly portable and easy to install. Additionally, XTP itself supports general communications needs which satisfies the diverse requirements of the military. Finally, SandiaXTP supports the ability to send both reliable and unreliable datagrams by multicast or unicast. This is particularly important, since not all military message traffic requires assured delivery of data.

IV. EXPERIMENTS

A. INTRODUCTION

In Chapter II, XTP was determined to be a protocol that seems to fulfill the needs of the Naval Amphibious Readiness Group scenario. In Chapter III, the operating environment was described in which experiments would be conducted to determine the ability of XTP to reliably transfer data to single and multiple receivers concurrently. This chapter is dedicated to the description, summarization and graphical representation of those experiments. First, a description of the experiments is presented; this is followed by expectations of protocol fortuity, and finally, actual results.

B. DESCRIPTION OF EXPERIMENTS

Any protocol which is to be adopted for use in a military application must exhibit both scalability and survivability. For XTP to accomplish these goals, it must be able to effectively transfer messages consisting of a varying *amount* of data (i.e., small and large messages) via unicast and multicast (to increasing numbers of receivers) under varying error conditions. For this thesis, there will be 32 experiments, 16 of which will consist of the transmission of a small message, and 16 for a large message. Each of these 16 experiments will consist of a 4 x 4 matrix made up of four transmission methods (unicast, and multicast with one, two, and three receivers), and four levels of induced errors (none, one in 25 packets, one in ten, and one in five).

1. Small Message (50 Kilobytes)

The transmission of a 50 kilobyte message is intended to simulate the text-only data transfer between the Naval Beach Group and forces afloat. This message should include descriptions of the landing zone including surf, weather, terrain, etc., as well as the status of both personnel and supplies.

2. Large Message (One Megabyte)

The one megabyte message is the simulation of sending a graphical image of the landing zone to the forces afloat. Frequent transmission of a single, comprehensive "snapshot" of the landing zone ensures that all concerned parties are in possession of a common tactical picture as soon as it becomes available.

3. Transmission Method

XTP provides both a unicast and multicast transmission service. Unicast may be used whenever only one recipient is to be addressed, while multicast is used for groups of one or more recipients. For these experiments, the multicast transmissions are sent to one, two, and three receivers.

4. Induced Errors

Under any military application, a hostile communications environment is not only possible, it must be considered normal. Therefore, errors will be introduced into the data transfer stream by choosing one receiver to drop packets. To effectively evaluate the performance of the transport protocol, baseline measurements (i.e., no induced errors) must first be taken, followed by performance measurements taken under various (simulated) conditions of link effectiveness and transfer path noise. Therefore, errors will be introduced with means of one packet per 25 and one per 10, with the final examination simulating an extremely hostile communications environment and/or a very lossy receiver, losing on average one of every five packets. The packet loss is distributed uniformly about the mean.

5. Buffer Size

Another parameter in the experiments is the size of the *buffer*. This is an amount of reserved space in both the transmitter and receiver(s) that stores the data being transferred. For these experiments, the buffer size will be varied from 64 bytes to 8192 bytes, increasing by powers of two (i.e., 2^6 bytes, 2^7 bytes, 2^8 bytes, etc.). The importance of the buffer is in the method of data transfer: the program will effect a repetitive loop sending an amount of data equal to the size of the buffer to the receiver(s) until all data has been sent and acknowledged by the receivers. Thus larger buffer sizes cause fewer calls to XTP to transfer the same amount of data.

6. Number of Iterations

With a knowledge of all the variables that influence protocol performance, it must now be noted that a single transmission of data will not consistently provide results characteristic of system capabilities. For this reason, three sets of data will be collected for each experiment, and the numbers will then be averaged to provide a more trustworthy basis for analysis.

C. EXPECTED RESULTS

1. Introduction

As with any set of experiments, some predictions can be made to form a hypothesis of performance. In this case, all variables must be considered – one at a time – and then be compared to actual results.

During the experiments, the "metric" test program (supplied with the SandiaXTP distribution) will be utilized. This will yield three values for each successful iteration:

- Timing – the amount of time consumed in the end-to-end delivery of the message (in milliseconds).
- Throughput – the measure of the rate of data delivery from end to end (in megabits per second).
- Latency – the time to deliver a message from end to end (in milliseconds per call).

Since the most useful performance characteristic of a transport protocol is throughput, all performance results will be viewed in those terms.

2. Unicast Versus Multicast

Unicast is expected to yield higher throughput than multicast as multicast incurs overhead to manage the group of receivers. This is because XTP builds an auxiliary context (protocol control block) for each receiver in a multicast group (Strayer, 1997).

3. Increasing the Number of Multicast Receivers

Increasing the number of multicast receivers should result in lower throughput, as each receiver must provide the requisite acknowledgments prior to the transmitter moving the send window. This means that the multicast throughput is highly dependent upon the performance of the slowest or most error prone receiver. If a receiver that demonstrates performance faster than, or equal in speed to the existing receivers is added, throughput should slow very slightly, as one additional acknowledgment must make it back to the transmitter prior to the packet transfer. However, if a slower receiver is added, throughput should show a more pronounced reduction.

4. Induction of Errors

The instance of errors in the transmission of data necessitates the retransmission of packets and, therefore, a baseline experiment is expected to have the highest throughput, while the introduction of one error per 25 packets should reduce the throughput by approximately 4% (one in 25). By similar argument, one error in ten packets should reduce the baseline throughput by 10%, and one error in five packets should degrade performance by 20%. Overall throughput will be determined by the slowest receiver.

5. Buffer Size

The buffer determines how much data may be transmitted in a single call to XTP. Since the buffer size is increased by powers of 2, the throughput should also increase exponentially, resulting in a linear curve. The degree of increase should be approximately double, until reaching the maximum XTP data transfer limitation of 1448 bytes per packet (determined by a 1500 byte Ethernet frame minus 20 bytes for an IP header and 32 bytes for the XTP header). When transferring packets larger than 1448 bytes, there is mandatory packet segmentation and re-assembly by the transmitter and receiver(s) respectively. This means that a reduction in throughput should occur for a buffer size slightly larger than 1448 bytes since two packets are now required. There will be an amortization of this effect for further increases above 1448, as well as higher multiples of 1448, as the effects of increasing overhead are diminished. Similarly, as the amount of data to be sent is doubled, throughput should approximate a doubling minus the sum of the costs of segmentation and re-assembly, and incurred overhead.

D. ACTUAL RESULTS ANALYSIS (VARYING TRANSMISSION METHOD)

The tables presented in this chapter depict the averages of the three iterations of each experiment (Appendix A shows each of the tables of results from the three sets of data drawn from the experiments.). The analyses that follow relate buffer size and throughput for each experiment. The reason for this presentation is to demonstrate the *scalability* of XTP: how the addition of receivers (with all other parameters held constant) affects throughput.

The first four sets of results examine the behavior of the implementation and protocol during the transmission of a 50 kilobyte (51200 byte) message with and without induced errors. The following four sets examine performance during transmission of a 1 megabyte (1024000 byte) message. For each of these groups of four, the initial investigation is conducted in an environment of zero induced errors. The second set of data is the result of an induced error count of one packet per every 25. The third increases the induced error count to one per ten packets, and finally, the noisy channel/lossy receiver possibility is presented as the number of lost packets reaches one in five.

1. 50 Kilobyte Message Size

a. *No induced Errors*

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.126	0.168	0.202	0.198
128	0.284	0.221	0.352	0.374
256	0.495	0.477	0.570	0.580
512	1.090	0.702	0.804	0.776
1024	1.702	0.916	1.032	1.021
2048	2.610	0.798	1.064	1.101
4096	3.735	1.114	1.194	1.215
8192	4.537	1.150	1.200	1.212

Table 4.1 Throughput of a 50 Kbyte Message Transmitted with No Induced Errors

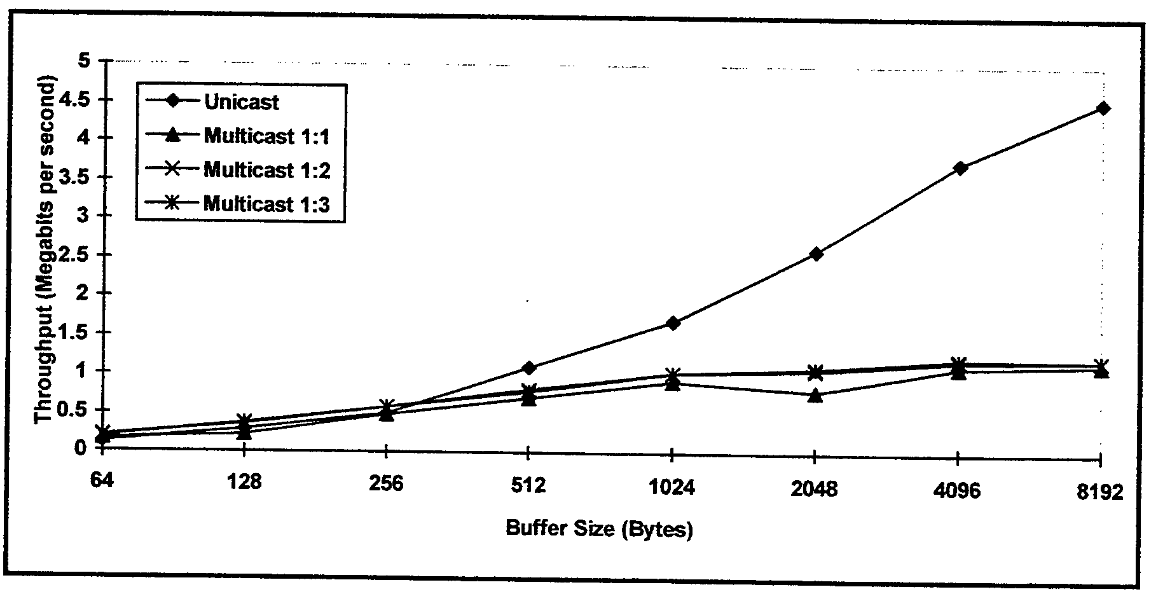

Figure 4.1 Throughput of a 50 Kbyte Message Transmitted with No Induced Errors

A message of this relatively small size is transmitted very quickly and therefore, small idiosyncrasies in topography state may yield unexpected results as demonstrated by the smaller unicast throughput at the 64 byte buffer size (Table 4.1). Above this level, however, the unicast results conform to expectations, exceeding those of the multicast.

The multicast transmission results demonstrate the overhead of managing a group of receivers. However, the addition of receivers did not significantly hamper throughput.

b. Induced errors: 1 of 25 packets

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.189	0.151	0.159	0.169
128	0.360	0.278	0.283	0.319
256	0.673	0.447	0.354	0.402
512	1.196	0.689	0.622	0.604
1024	2.272	0.912	0.867	0.904
2048	2.774	0.983	0.938	0.848
4096	4.479	1.085	1.209	1.108
8192	4.434	1.176	1.040	0.904

Table 4.2 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 25 Packets Dropped

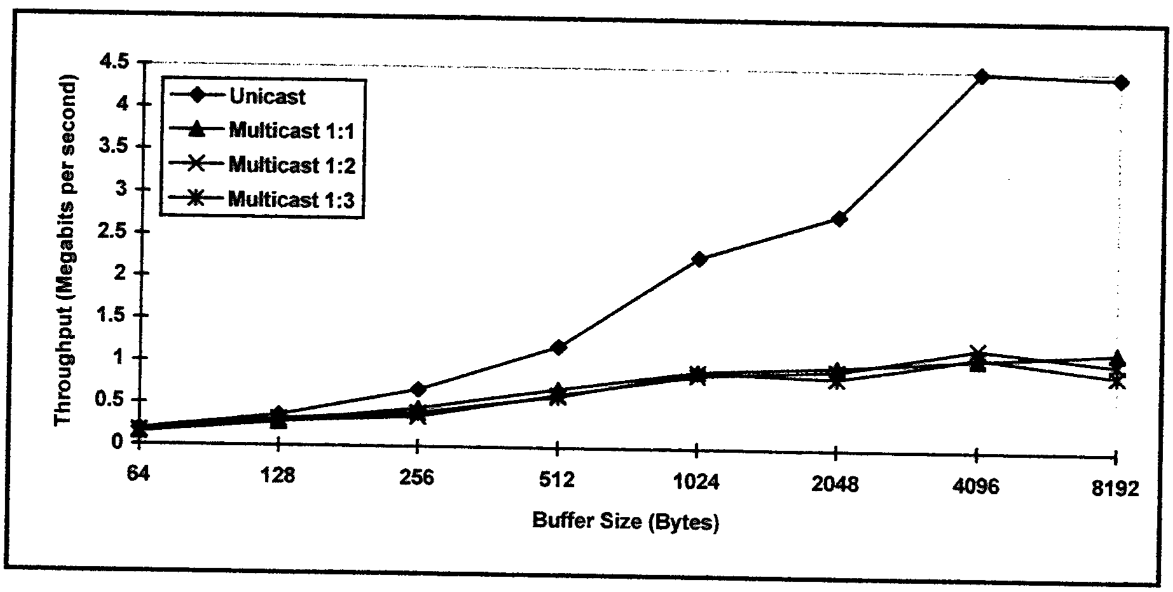

Figure 4.2 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 25 Packets Dropped

The results of the 50 Kbyte message with induced errors not only reflect a more realistic data transfer, they also conform more strongly to expected results. The unicast transmission retains the highest throughput for all buffer sizes, and exhibits an asymptotic increase up to and including the 1024 byte buffer level as a result of the log scale. This is consistent with expectations, resulting from the exponential increase in the buffer size. However, the reduction in throughput as the buffer is increased from 4096 bytes to 8192 bytes deserves attention: this anomaly may be explained by the FASTNAK switch being set, along with XTP's use of GO BACK N retransmission (as described in Chapter III). With a large buffer size, XTP's internal buffers store numerous packets (six for an 8192 byte buffer) for each call to XTP. When an error occurs, several packets may have to be retransmitted. This will result in a "leveling off" and possible reduction in throughput, as seen when the buffer is increased from 4096 bytes to 8192 bytes. This effect is observed in Figure 4.2, as well as on future graphs at the same buffer size transition.

The multicast results show relatively constant throughput for an increasing number of receivers at any buffer size, which is a desired trait of a multicast protocol: scalability. Increasing the number of receivers does not significantly detract from system throughput.

c. *Induced errors: 1 of 10 packets*

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.120	0.153	0.152	0.131
128	0.244	0.275	0.237	0.231
256	0.446	0.482	0.387	0.386
512	0.772	0.625	0.598	0.573
1024	1.402	0.903	0.659	0.713
2048	2.156	0.520	0.706	0.704
4096	2.971	[1]	0.742	0.806
8192	2.551	[1]	0.749	0.747

[1] No successful runs were completed at this buffer size.

Table 4.3 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 10 Packets Dropped

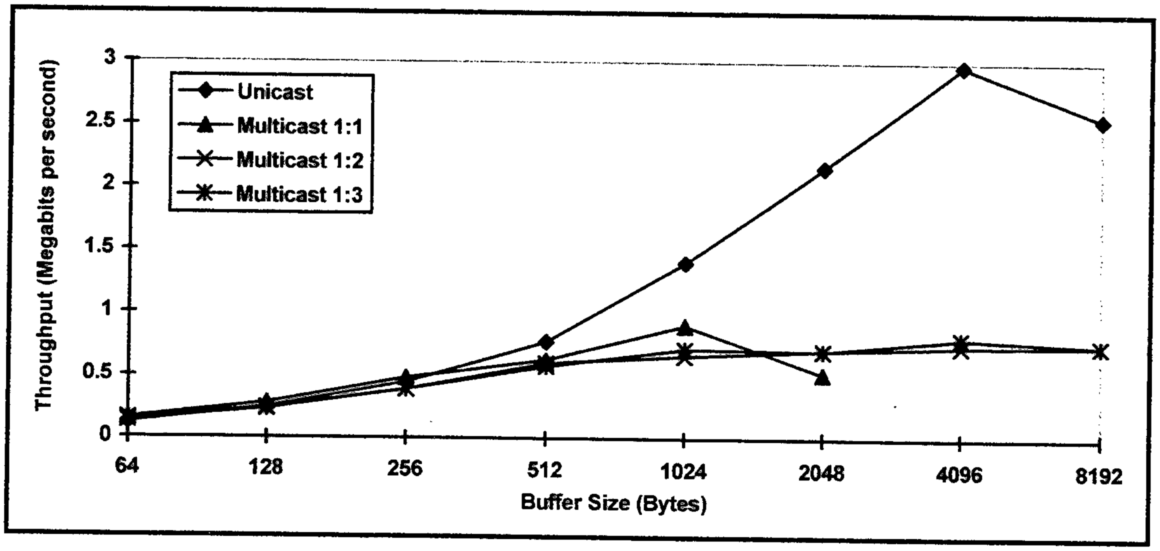

Figure 4.3 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 10 Packets Dropped

Unicast performs as expected, demonstrating the reduction in throughput as the buffer is increased from 4096 to 8192 bytes as per the explanation in the previous subsection. The 1024 byte buffer demonstrates (roughly) expected results with multicast performance remaining below that of unicast, but at a degradation level of less than 50%. Proceeding on to the 2048 byte level, the multicast (one receiver) transmission shows unexpected behavior by exhibiting a significant reduction in throughput before failing to complete, while the multicast (two receivers) and multicast (three receivers) transmission paths demonstrate a slightly reduced – yet relatively constant – throughput.

Failures due to timeouts can be attributed to a timing algorithm within XTP which monitors transmission time. In the face of errors, the algorithm does not know whether the channel requires a longer round trip time (RTT) due to a longer route, or if packets were lost, necessitating retransmission. If the wait timer (WTIMER) expires without a response from the receiver(s), the timer backs off exponentially and is restarted to allow sufficient time for message delivery. However, if several back-to-back errors occur, the timer backs off for a sufficiently long time, allowing an ancillary (watchdog) timer (CTIMEOUT) to expire. This is believed to be the cause of the failed experiments to multiple receivers, as the addition of receivers slows the message transfer to the speed of the slowest receiver.

d. Induced errors: 1 of 5 packets

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.168	0.131	0.134	0.126
128	0.305	0.230	0.233	0.216
256	0.566	0.407	0.349	0.372
512	0.974	0.510	0.494	0.533
1024	1.114	0.651	0.638	0.718
2048	1.186	0.599	0.445	0.679[1]
4096	1.945	0.765	[2]	[2]
8192	0.714	0.779	[2]	[2]

[1] Average compiled with only one set of input data.
[2] No successful runs were completed at this buffer size.

Table 4.4 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 5 Packets Dropped

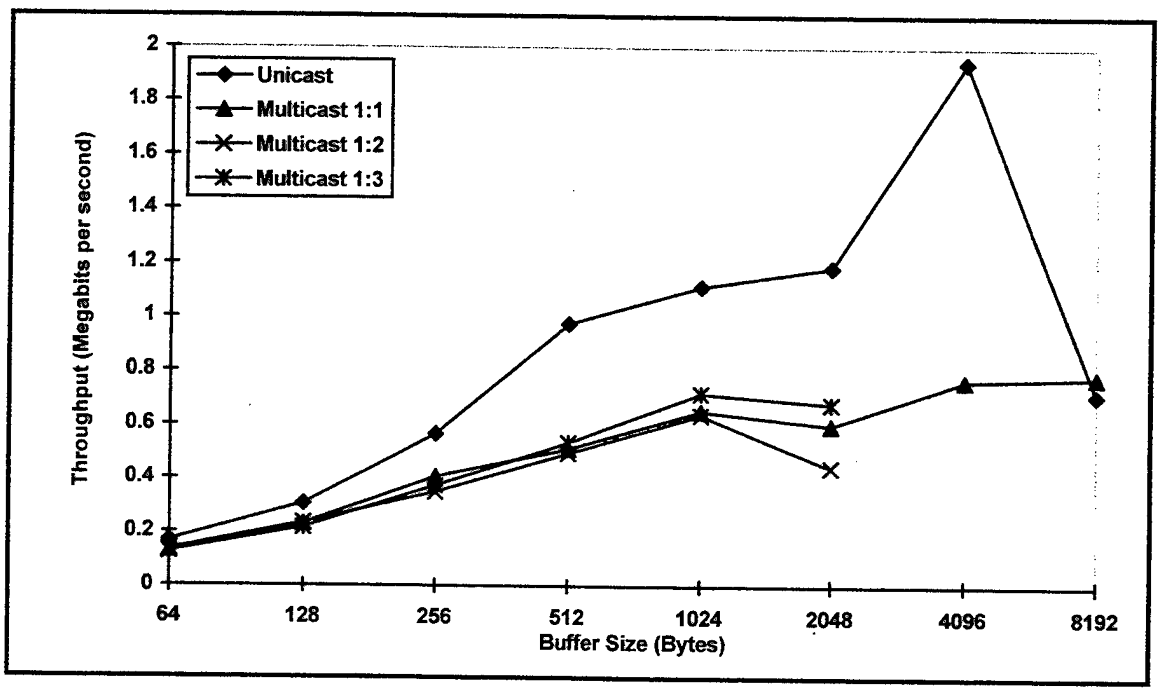

Figure 4.4 Throughput of a 50 Kbyte Message Transmitted with Induced Errors: 1 of 5 Packets Dropped

The continuing pattern of increasing throughput with unicast leading multicast is shown in Figure 4.4. Performance above the 1024 byte buffer becomes erratic due to the segmentation and re-assembly process described in Chapter III, with a cessation of success above 2048 bytes for multicast to two and three receivers due to the timeout algorithm.

2. One Megabyte Message Size

a. No induced Errors

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.108	0.104	0.105	0.105
128	0.216	0.215	0.210	0.211
256	0.435	0.417	0.447	0.445
512	0.850	0.884	0.886	0.887
1024	1.500	1.489	1.469	1.470
2048	2.202	2.190	2.294	2.261
4096	3.629	3.729	3.711	3.648
8192	3.522	5.091	5.210	4.991

Table 4.5 Throughput of a One Mbyte Message Transmitted with No Induced Errors

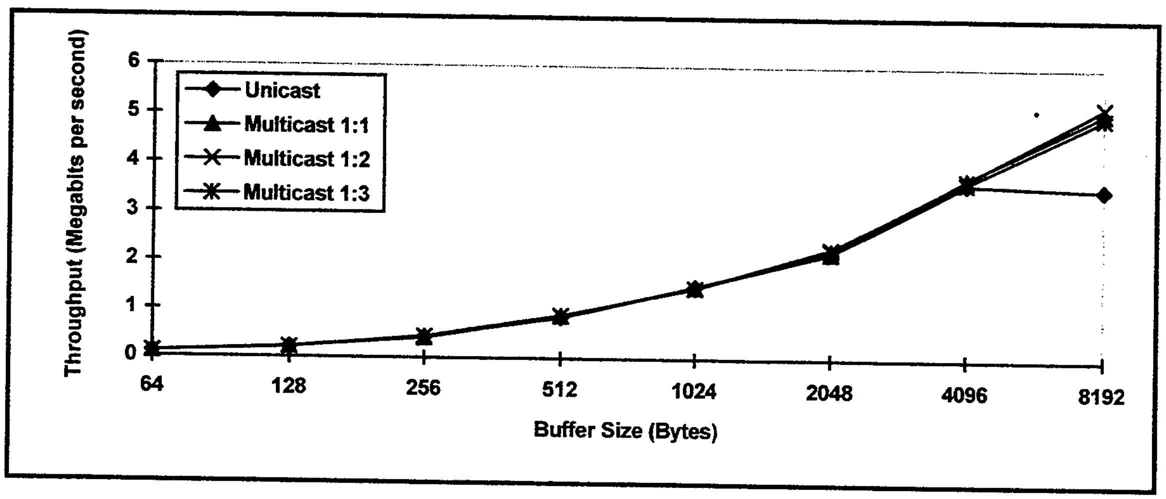

Figure 4.5 Throughput of a One Mbyte Message Transmitted with No Induced Errors

Transmission of a one megabyte message allows for a more realistic evaluation of protocol performance, as sporadic interference introduced by noisy channels does not have such a magnified effect on throughput. This is because the increased message size reduces the impact of periodic obstructions to data transfer. Figure 4.5 demonstrates this.

With this larger message size, all results conform to expectations, with one exception: the reduction in unicast performance when increasing the buffer size to 8192 bytes. This anomaly cannot be readily explained.

b. Induced errors: 1 of 25 packets

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.161	0.118	0.089	0.100
128	0.289	0.316	0.177	0.207
256	0.572	0.610	0.383	0.419
512	1.116	1.167	0.673	0.668
1024	2.025	1.648	1.071	1.080
2048	3.161	1.515[1]	1.871	2.041
4096	3.556	2.140	[3]	[3]
8192	3.607	2.622[2]	[3]	[3]

[1] Average compiled with only one set of input data.
[2] Average compiled with only two sets of input data
[3] No successful runs were completed at this buffer size.

Table 4.6 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 25 Packets Dropped

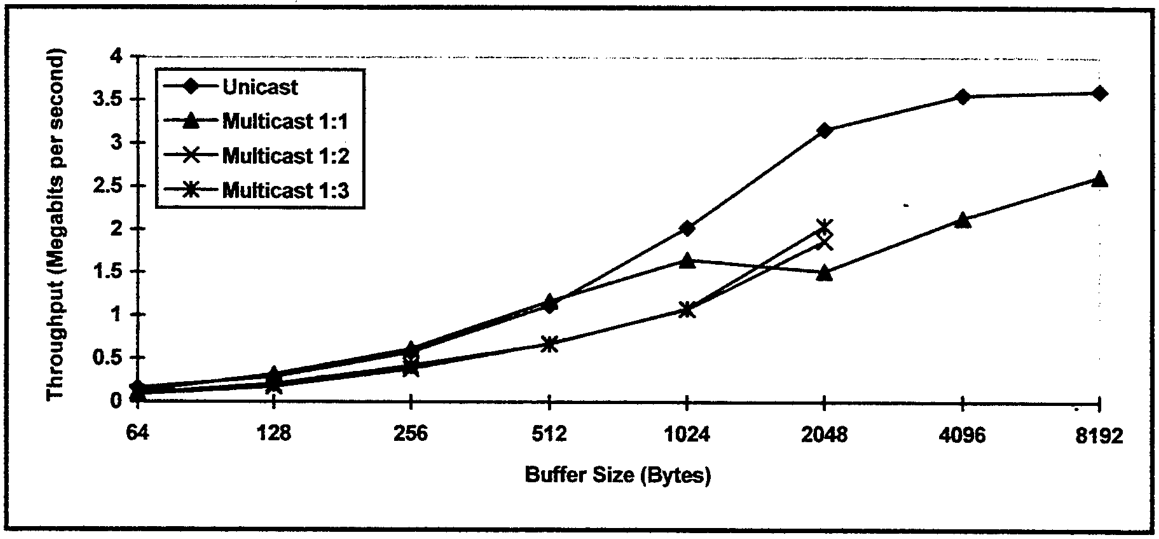

Figure 4.6 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 25 Packets Dropped

Figure 4.6 shows the throughput of a one megabyte message transmitted with the induction of one in 25 errors. Unicast performance shows the expected exponential increase up to a buffer size of 1024 bytes, and a reduction in performance increases for subsequent increases in buffer size. This is due to the XTP 1448 byte packet size and the requirement for packet segmentation and re-assembly as discussed earlier.

Multicast throughput is somewhat less, with one item of note: at the 2048 byte buffer level with transmission to one receiver, an unevenness in the performance graph is due to only one successful run at these parameters. With only one set of data to rely on, the effects of averaging three runs is eliminated, and less trustworthy data is provided as input to the graph. In this case, that single successful run yielded slower than expected throughput.

c. *Induced errors: 1 of 10 packets*

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.108	0.033	0.061	0.087
128	0.222	0.075	0.152	0.182
256	0.429	0.327	0.311	0.365
512	0.809	0.657	0.730	0.710
1024	1.422	1.015	1.226	1.211
2048	1.892	1.504	1.408	[3]
4096	2.374	1.531	1.770	[3]
8192	1.447	2.124[1]	1.748[2]	[3]

[1] Average compiled with only one set of input data.
[2] Average compiled with only two sets of input data
[3] No successful runs were completed at this buffer size.

Table 4.7 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 10 Packets Dropped

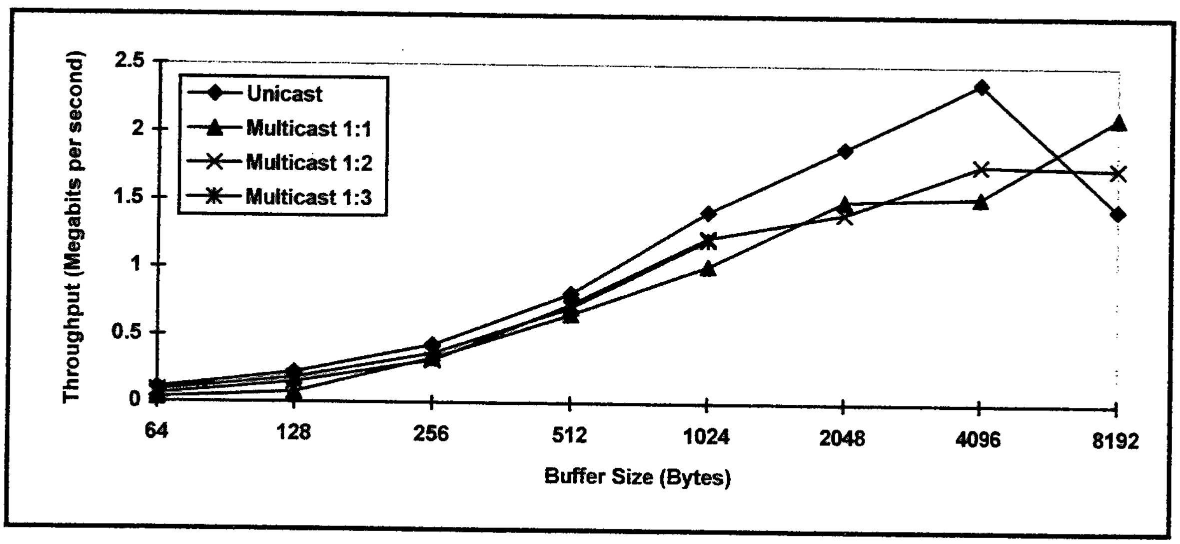

Figure 4.7 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 10 Packets Dropped

Figure 4.7 shows the increasing degradation of throughput as the number of induced errors is increased to one in ten packets. All results are to be expected, remembering that the Ethernet limitation of packet size hampers performance increases beyond the 1024 byte buffer, and that the watchdog timer expiration may result in an unsuccessful experiment run. This is the case in multicasting to three receivers with a buffer size larger than 1024 bytes.

d. *Induced errors: 1 of 5 packets*

Buffer Size (Bytes)	Unicast	Multicast 1:1	Multicast 1:2	Multicast 1:3
64	0.107	0.107	0.052	0.085
128	0.222	0.194	0.153	0.205
256	0.420	0.372	0.318	0.348
512	0.775	0.644	0.709	0.684
1024	1.332	1.060	0.945	1.033
2048	0.989	1.223	0.970	[2]
4096	0.267	1.260[1]	[2]	[2]
8192	0.541	[2]	[2]	[2]

[1] Average compiled with only two sets of input data.
[2] No successful runs were completed at this buffer size.

Table 4.8 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 5 Packets Dropped

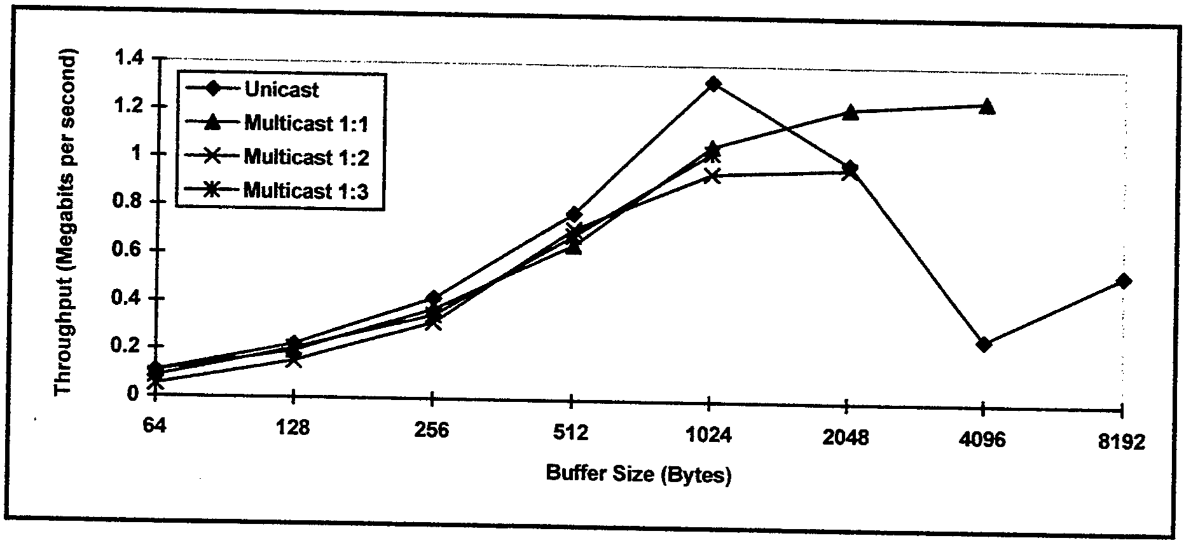

Figure 4.8 Throughput of a One Mbyte Message Transmitted with Induced Errors: 1 of 5 Packets Dropped

The increase in induced errors to one in five yields the graph depicted in Figure 4.8. Once again, the results up to and including the buffer size of 1024 bytes are in line with expectations. Coincidentally, it is beyond this 1 kilobyte level that successful multicast transmission to three receivers is lost. Multicasting to two receivers is lost beyond the next larger buffer size (2048 bytes), with multicast to one receiver failing for the 8192 byte buffer size. This phenomenon is explained by the combination of the required retransmissions of large size packets consuming sufficient amounts of time to activate the watchdog timer shutdown mechanism.

While unicast transmission operates successfully throughout the range of buffer sizes, considerable degradation occurs beyond the 1024 byte buffer, as with multicast (as a whole).

3. Transmission Method Analysis Summary

Generally, the figures above depict expected protocol performance up to and including a buffer size of one kilobyte. Beyond this, the Ethernet packet size limitation of 1448 bytes obligates packet segmentation by the transmitter, and re-assembly by the receiver(s). This, combined with the introduction of induced errors and the resulting watchdog timer influence, results in negligible – and sometimes negative – performance increases at larger buffer sizes. XTP was shown to be scalable up to three receivers as there is no appreciable change in throughput as the number of receivers is increased.

E. ACTUAL RESULTS ANALYSIS (VARYING INDUCED ERRORS)

The previous subsection concentrated on an analysis of throughput variations as the transmission method shifted from unicast to multicast, and the subsequent addition of receivers to the multicast transfer of data. This was done to examine the scalability of

XTP. Now the effect of the inclusion of an increasing amount of errors on a given transmission method is examined. This analysis is intended to demonstrate the survivability of the protocol.

1. 50 Kilobyte Message Size

a. Unicast

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.126	0.189	0.120	0.168
128	0.284	0.360	0.244	0.305
256	0.495	0.673	0.446	0.566
512	1.090	1.196	0.772	0.974
1024	1.702	2.272	1.402	1.114
2048	2.610	2.774	2.156	1.186
4096	3.735	4.479	2.971	1.945
8192	4.537	4.434	2.551	0.714

Table 4.9 Throughput of a 50 Kbyte Unicast Message with Varying Induced Errors

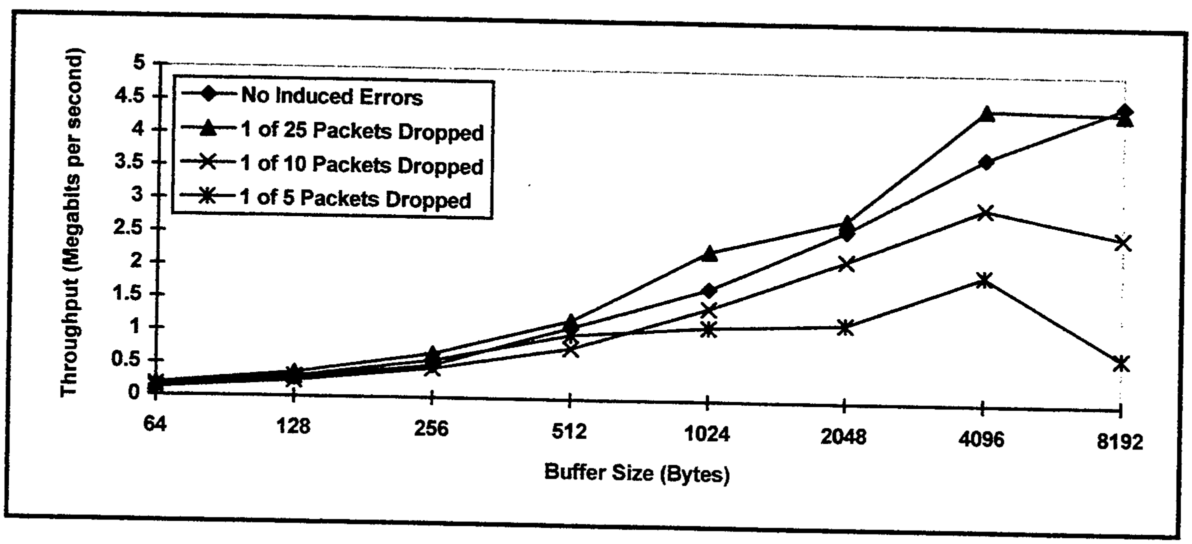

Figure 4.9 Throughput of a 50 Kbyte Unicast Message with Varying Induced Errors

Figure 4.9 shows the effects of inducing errors into a unicast transmission of a small (50 kilobyte) message. While it is expected that increasing the amount of errors (from one in 25, to one in ten, to one in five) will result in a slower throughput, as is the case; it is surprising that throughput *rose* in the transition from "no induced errors" to "one in 25 packets". This can be explained by the transmission of control packets in a FASTNAK scheme ironically *raising* throughput.

The prompt sending of a control packet by a receiver signaling an error (and resulting in packet retransmission) interrupts the flow of data packets by the transmitter and initiates the GO BACK N retransmission scheme. With buffer sizes less than 1448 bytes, the retransmitted data may be piggy-backed onto previously scheduled packets (filling in the available space in each packet). By filling the available space within each packet to a greater degree, throughput does not suffer, as shown in Figure 4.9. This phenomenon does not show increased throughput beyond the initial error level of one packet per 25, as the ramp up to one in ten yields a greater effect of timing delay and the inherent reduction in throughput due to the GO BACK N retransmission scheme in the face of many errors.

b. Multicast to One Receiver

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.168	0.151	0.153	0.131
128	0.221	0.278	0.275	0.230
256	0.477	0.447	0.482	0.407
512	0.702	0.689	0.625	0.510
1024	0.916	0.912	0.903	0.651
2048	0.798	0.983	0.520	0.599
4096	1.114	1.085	[1]	0.765
8192	1.150	1.176	[1]	0.779

[1] No successful runs were completed at this buffer size.

Table 4.10 Throughput of a 50 Kbyte Multicast (One Receiver) Message with Varying Induced Errors

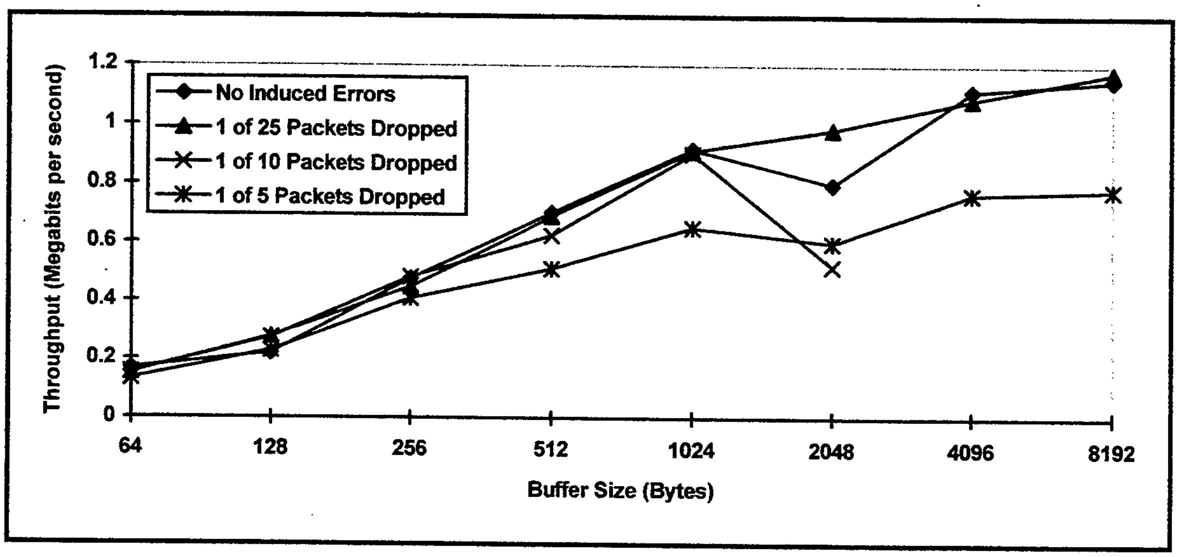

Figure 4.10 Throughput of a 50 Kbyte Multicast (One Receiver) Message with Varying Induced Errors

Figure 4.10 shows the effects of varying the number of induced errors on a multicast transmission to one receiver. While the disparity between "no errors" and "one in 25" is greatly reduced, there is an unusual termination of successful experimentation above a buffer size of 2048 bytes with one error in ten packets, as noted earlier.

c. *Multicast to Two Receivers*

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.202	0.159	0.152	0.134
128	0.352	0.283	0.237	0.233
256	0.570	0.354	0.387	0.349
512	0.804	0.622	0.598	0.494
1024	1.032	0.867	0.659	0.638
2048	1.064	0.938	0.706	0.445
4096	1.194	1.209	0.742	[1]
8192	1.200	1.040	0.749	[1]

[1] No successful runs were completed at this buffer size.

Table 4.11 Throughput of a 50 Kbyte Multicast (Two Receivers) Message with Varying Induced Errors

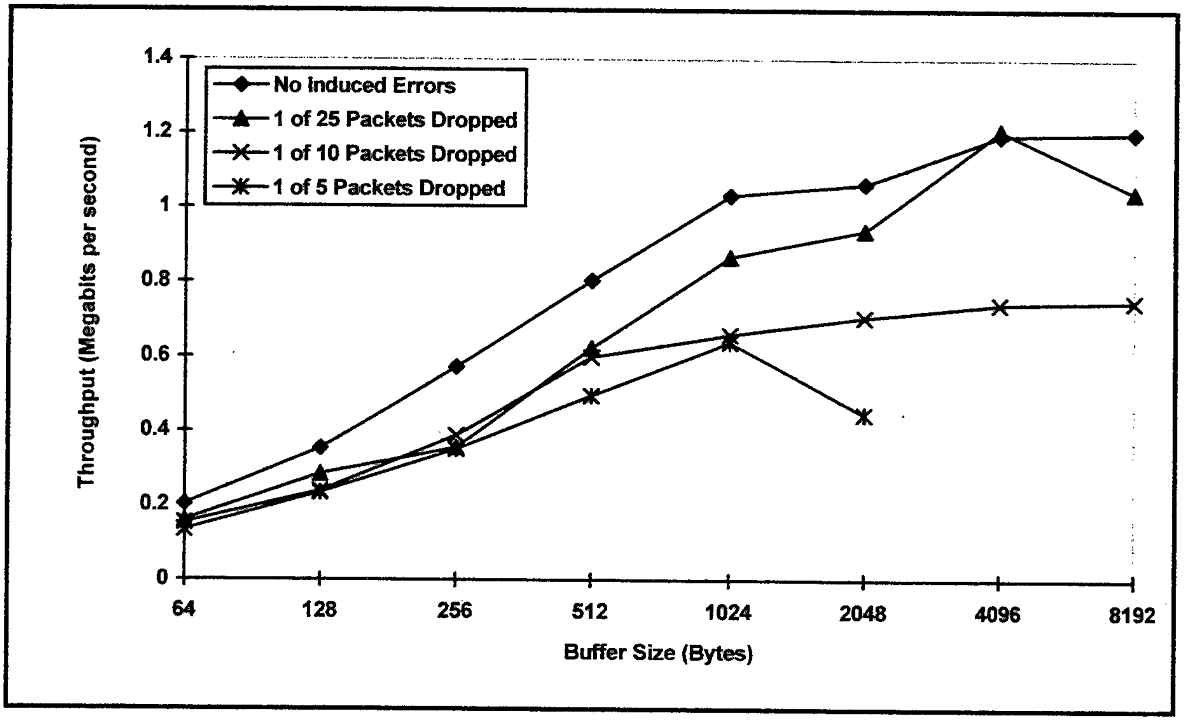

Figure 4.11 Throughput of a 50 Kbyte Multicast (Two Receivers) Message with Varying Induced Errors

Figure 4.11 demonstrates expected results: throughput is reduced as the number of errors is increased, and the single instance of unsuccessful transmission occurs at the highest error rate, and at the higher buffer levels. This is due to expiration of the watchdog timer.

d. Multicast to Three Receivers

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.198	0.169	0.131	0.126
128	0.374	0.319	0.231	0.216
256	0.580	0.402	0.386	0.372
512	0.776	0.604	0.573	0.533
1024	1.021	0.904	0.713	0.718
2048	1.101	0.848	0.704	0.679[1]
4096	1.215	1.108	0.806	[2]
8192	1.212	0.904	0.747	[2]

[1] Average compiled with only one set of input data.
[2] No successful runs were completed at this buffer size.

Table 4.12 Throughput of a 50 Kbyte Multicast (Three Receivers) Message with Varying Induced Errors

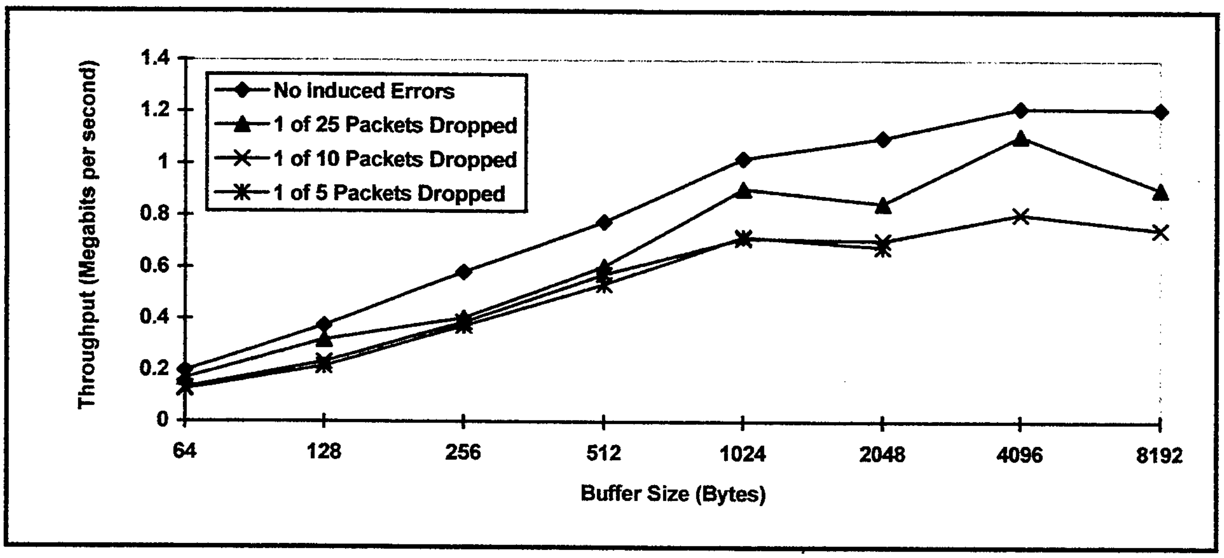

Figure 4.12 Throughput of a 50 Kbyte Multicast (Three Receivers) Message with Varying Induced Errors

The multicast transmission to three receivers depicted in Figure 4.12 conforms to expectations. Worthy of note is the similarity to Figure 4.11 both visually and numerically: performance remains stable with the addition of another receiver with no appreciable change in throughput. This indicates scalability.

2. One Megabyte Message Size

a. Unicast

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.108	0.161	0.108	0.107
128	0.216	0.289	0.222	0.222
256	0.435	0.572	0.429	0.420
512	0.850	1.116	0.809	0.775
1024	1.500	2.025	1.422	1.332
2048	2.202	3.161	1.892	0.989
4096	3.629	3.556	2.374	0.267
8192	3.522	3.607	1.447	0.541

Table 4.13 Throughput of a One Mbyte Unicast Message with Varying Induced Errors

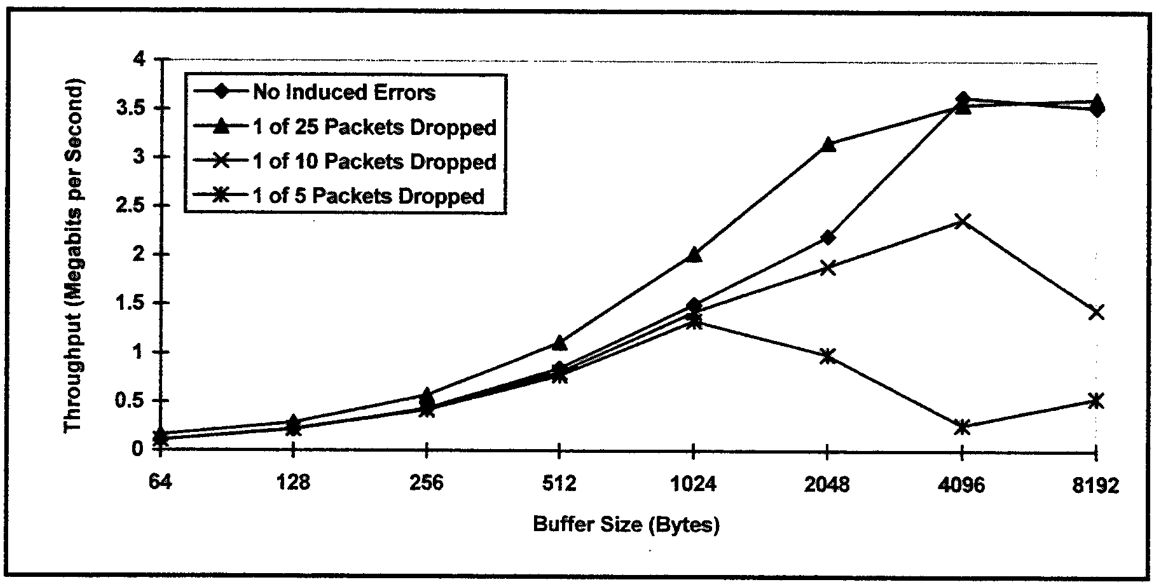

Figure 4.13 Throughput of a One Mbyte Unicast Message with Varying Induced Errors

Figure 4.13 shows the unicast transfer of a one megabyte message. This larger message size acts to stabilize inconsistencies prevalent in noisy channels and the effects of errors on small messages. Highly predictable results are seen at buffer sizes up to and including 1024 bytes, with continuing predictable data in all curves, except for the worst case of one lost packet in five. The severe loss in throughput is attributable to the reduction in throughput resulting from many retransmissions.

b. Multicast to One Receiver

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.104	0.118	0.033	0.107
128	0.215	0.316	0.075	0.194
256	0.417	0.610	0.327	0.372
512	0.884	1.167	0.657	0.644
1024	1.489	1.648	1.015	1.060
2048	2.190	1.515[1]	1.504	1.223
4096	3.729	2.140	1.531	1.260[2]
8192	5.091	2.622[2]	2.124[1]	[3]

[1] Average compiled with only one set of input data
[2] Average compiled with only two sets of input data
[3] No successful runs were completed at this buffer size.

Table 4.14 Throughput of a One Mbyte Multicast (One Receiver) Message with Varying Induced Errors

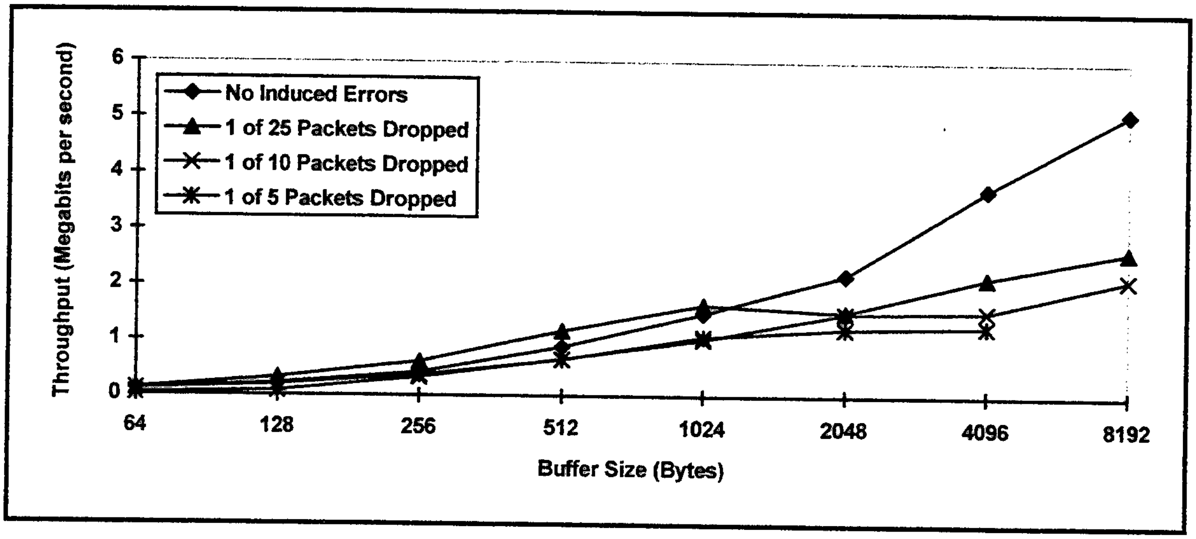

Figure 4.14 Throughput of a One Mbyte Multicast (One Receiver) Message with Varying Induced Errors

The shift to a multicast transmission (independent of the number of receivers) has smoothed protocol performance and produced expected results The introduction of errors results in a decrease in throughput, and there is an eventual attainment of non-successful experimentation under worst case conditions.

c. *Multicast to Two Receivers*

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.105	0.089	0.061	0.052
128	0.210	0.177	0.152	0.153
256	0.447	0.383	0.311	0.318
512	0.886	0.673	0.730	0.709
1024	1.469	1.071	1.226	0.945
2048	2.294	1.871	1.408	0.970
4096	3.711	[2]	1.770	[2]
8192	5.210	[2]	1.748[1]	[2]

[1] Average compiled with only two sets of input data.
[2] No successful runs were completed at this buffer size.

Table 4.15 Throughput of a One Mbyte Multicast (Two Receivers) Message with Varying Induced Errors

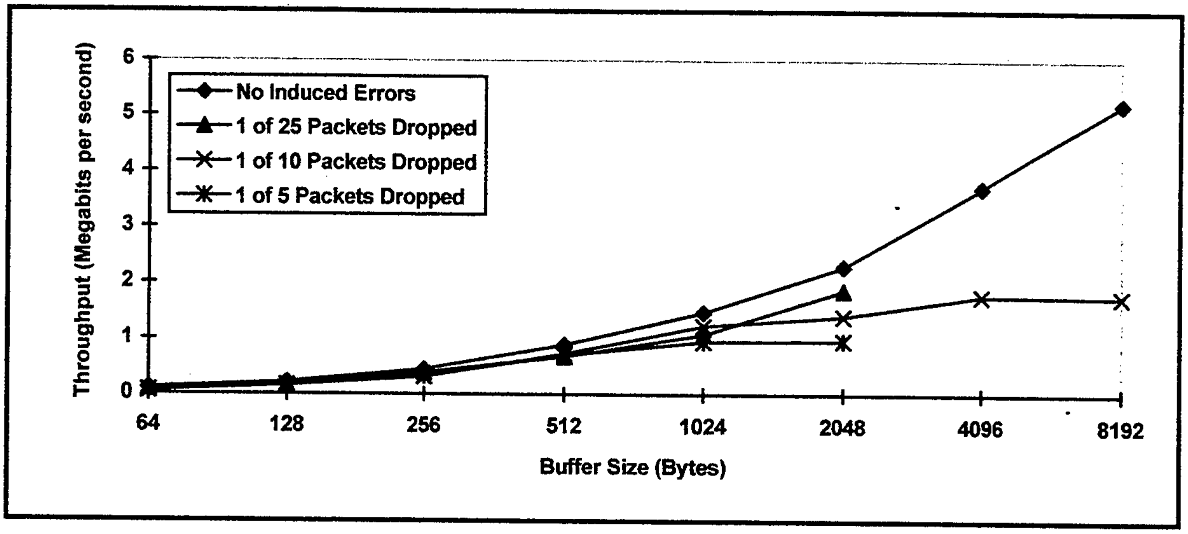

Figure 4.15 Throughput of a One Mbyte Multicast (Two Receivers) Message with Varying Induced Errors

Figure 4.15 shows a multicast transmission to two receivers. Other than the unexpected halting of experiment success beyond 2048 bytes for both one of 25 errors and one in five errors, results are predictable.

d. Multicast to Three Receivers

Buffer Size (Bytes)	No Induced Errors	1 of 25 Packets Dropped	1 of 10 Packets Dropped	1 of 5 Packets Dropped
64	0.105	0.100	0.087	0.085
128	0.211	0.207	0.182	0.205
256	0.445	0.419	0.365	0.348
512	0.887	0.668	0.710	0.684
1024	1.470	1.080	1.211	1.033
2048	2.261	2.041	[1]	[1]
4096	3.648	[1]	[1]	[1]
8192	4.991	[1]	[1]	[1]

[1] No successful runs were completed at this buffer size.

Table 4.16 Throughput of a One Mbyte Multicast (Three Receivers) Message with Varying Induced Errors

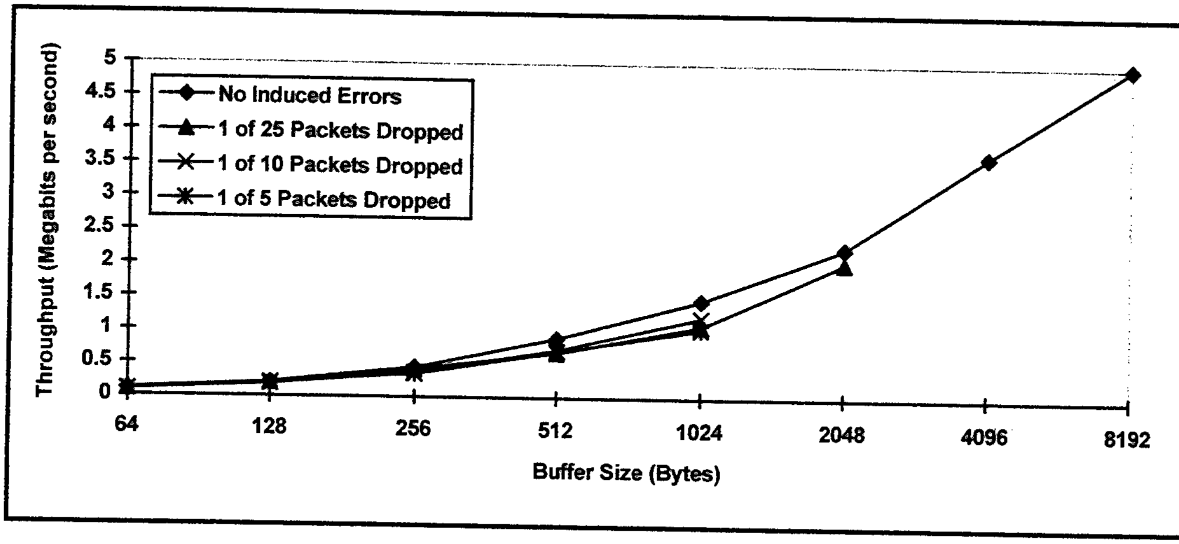

Figure 4.16 Throughput of a One Mbyte Multicast (Three Receivers) Message with Varying Induced Errors

The survivability of XTP is shown best in Figure 4.16. This graph depicts a graduation to the maximum number of induced errors while multicasting to the maximum number of receivers (three). Throughput remains relatively constant in the face of increasing taxation on protocol retransmission capabilities.

3. Variation of Induced Errors Analysis Summary

In general, the figures in this subsection conform to expectations by depicting a reduction in throughput as the number of induced errors is increased. The survivability of XTP is shown by repetitively high – and relatively stable -- throughput levels (approximately one megabit per second for the large message) in the face of increasing errors. However, timer settings impeded the satisfactory completion of transmissions which combined high error rates with multicasting. Additionally, performance above a buffer size of 4096 bytes became extremely untrustworthy and included numerous experiment failures, due to segmentation and re-assembly combined with an increasing numbers of errors.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

1. Viability of Reliable Multicast

The military needs reliable multicast because in many circumstances (e.g., a tactical theater), a single transmitter needs to send information to many destinations under both hostile conditions and conditions of limited bandwidth. It is also important that all participating units have the same information as soon as it becomes available. Unlike unicast, which is a point-to-point data transfer method, multicast provides data transfer capabilities to numerous receivers simultaneously. The results of the experiments conducted show that data can be transmitted to multiple receivers under dynamic buffer and error conditions while maintaining a relatively consistent throughput.

2. Reliable Multicast Concerns

While running experiments, there was one significant anomaly noted: data transfer at buffer sizes above one kilobyte were frequently unpredictable and periodically unsuccessful. This was most likely due to the Ethernet connection which restricted packet size to 1448 bytes. Buffer sizes above this mandate a segmentation and re-assembly process which hampers throughput and could foster timeout errors. On the other hand, these errors may have resulted from the testbed that was subject to workstation loading.

a. *Wait Timers*

When a packet is sent with the SREQ bit set, a wait timer (WTIMER) is started. If the transmitter does not receive an acknowledgment from the receiver(s) within the specified time after sending a packet, the timer will expire, starting a synchronizing handshake. In the synchronizing handshake the transmitter sends a control packet with SREQ set, and restarts the WTIMER. No data can be sent during the synchronizing handshake. The synchronizing handshake is guarded by an ancillary timer (CTIMEOUT), set in SandiaXTP to be 60 seconds. Completion failures were caused by the expiration of the CTIMEOUT during conducted experiments.

XTP uses lost packets to reset a Round Trip Timer (RTT). Because the actual routing pathway varies due to dynamic changes in the network, packet delivery times will vary as well. Since this only occurs under conditions where the route changes, the implementation of the backoff timer with the existing testbed topology was more harmful than helpful. This is simply because packets never left the gateway and a reroute was not possible.

These failures are disturbing and unacceptable in a tactical environment, so further testing must be conducted. However, it should be noted that this phenomenon could be solved by either setting the CTIMEOUT to larger values (20 minutes may be more appropriate), or by redesigning the synchronizing handshake to stop backing off exponentially once the values become large. Moreover, scalability also becomes an issue when considering wait timers, because more wait timers equates to greater overhead.

b. Go Back N and Buffer Sizes

When a retransmission is requested, the transmitter resends everything from the point of lost data on. The larger the buffer size, the more data – on average – to be retransmitted. In the presence of errors, there is a greater likelihood of the retransmitted data also getting lost when a lot of data is being retransmitted. This would cause further delay, and possibly set up conditions described above where the WTIMER grows large. The larger the WTIMER value, the more likely the CTIMEOUT will expire, aborting the connection. The use of a selective retransmission scheme could help reduce the effect of reduced throughput in lossy situations.

3. Testbed Concerns

The experiments described in Chapter IV were conducted on workstations in an operational network laboratory, with no restriction on use by others. The lack of a dedicated system *may* have increased experiment result variation due to the existing collision domain, although it is believed that this was not a major factor.

4. Summary

This thesis examined the feasibility of adopting XTP as a reliable multicast protocol to fulfill the needs of military combat data transfer. The protocol supports both unicast and multicast transmissions, with the addition of receivers producing a negligible impact on multicast throughput. Small and large files were successfully transferred to multiple receivers with throughputs in excess of one megabit per second. Occasional throughput levels exceeding five megabits per second were achieved, despite the Ethernet maximum data transfer rate of ten megabits per second.

XTP demonstrated survivability by maintaining successful data transfers to multiple receivers despite an increasing number of induced packet transfer errors, up to a tested level of 20% packet loss. Additionally, XTP is well suited to the requirement to adapt to a wide variety of hardware configurations. Because this variability of platforms is the norm in the military, the capability is essential in the reliable multicast protocol.

B. RECOMMENDATIONS

1. Expansion of this Study

The experiments conducted in support of this thesis were designed to verify the capability of XTP to successfully transfer data to multiple receivers under conditions of increasing errors. There are modifications to both the topology and the hardware that may be undertaken to further examine the potential for XTP to fulfill the requirements of a military reliable multicast protocol.

a. Remote Receivers

The experiments described in Chapter IV were restricted to a transmitter and receiver located within the same network segment. The inclusion of one or more receivers physically located at a site that requires routing via the Internet would allow examination of the protocol under imbalanced roundtrip times. The Internet also provides packet loss due to congestion on a dynamic scale that can not be predicted by operators, and therefore, performance measurements using this data pipe should be conducted.

b. Number of Receivers

Scalability is one of the most important features of a multicast protocol. While a three receiver scenario is sufficient for the tests presented in Chapter II, many military operations will involve more receivers. Increasing the number of receivers under given experiment conditions would yield valuable throughput observations and possibly reveal a maximum number of recipients.

c. Types of Receivers

Figure 3.2 showed the various platforms that are capable of running the SandiaXTP program. While this thesis utilized Sun SPARC4 workstations for the transmitter and all receivers, the use of a mix of compatible platforms (e.g., PC's, SGI's, etc.) would yield results indicative of operating with different military units.

2. Translation of this Study

a. Varying the Existing Experiments

XTP contains numerous tunable parameters, such as timers. Throughout the course of this study, it has been discovered that under stressful conditions, the settings of the timers may show a significant impact on data transmission results, especially for large files. For this reason, bulk data transfer should be done with more status requests during the normal course of the transmission, to help smooth the movement of the timer window. Making use of selective retransmission may also help here as well.

b. Use of Different Protocols

This thesis has shown that XTP is capable of fulfilling the military's needs for a reliable multicast protocol. Further research could compare the results in Chapter IV with the conduct of the same experiments using a different protocol. This is important in that any adopted protocol will most likely have attributes that come from several different protocols. However, it is important that XTP continues to be tracked as the specification matures so that further assessments can be made.

3. Fleet Implementation and Testing

Experiments conducted in a laboratory environment can provide very useful information to determine the applicability of prospective system acquisitions and significantly contribute to protocol enhancements. However, it is testing in the intended operating environment that will produce the most beneficial knowledge. The testbed for this study was created on a terrestrial topology; however, the intended operating environment based on the scenario will be a radio based topology. Therefore, as the protocol matures and deployment is possible, the protocol should be tested with a radio based topology in order to compare results.

APPENDIX - TABLES OF EXPERIMENT RESULTS

Note: Empty spaces indicate an unsuccessful run.

Unicast Transmission from One Transmitter to One Local Receiver
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3044	0.135	800	3.805
128	1214	0.337	400	3.035
256	899	0.456	200	4.495
512	427	0.959	100	4.270
1024	750	0.546	50	15.000
2048	168	2.438	25	6.720
4096	109	3.758	13	8.385
8192	104	3.938	7	14.857

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3387	0.121	800	4.234
128	1748	0.234	400	4.370
256	842	0.486	200	4.210
512	315	1.300	100	3.150
1024	224	1.829	50	4.480
2048	138	2.968	25	5.520
4096	139	2.947	13	10.692
8192	99	4.137	7	14.143

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3319	0.123	800	4.149
128	1463	0.280	400	3.658
256	756	0.542	200	3.780
512	405	1.011	100	4.050
1024	150	2.731	50	3.000
2048	169	2.424	25	6.760
4096	91	4.501	13	7.000
8192	74	5.535	7	10.571

Unicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 25 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2261	0.181	800	2.826
128	1117	0.367	400	2.792
256	652	0.628	200	3.260
512	364	1.125	100	3.640
1024	166	2.467	50	3.320
2048	136	3.012	25	5.440
4096	85	4.819	13	6.538
8192	104	3.938	7	14.857

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2137	0.192	800	2.671
128	1172	0.349	400	2.930
256	603	0.679	200	3.015
512	335	1.223	100	3.350
1024	229	1.789	50	4.580
2048	160	2.560	25	6.400
4096	102	4.016	13	7.846
8192	79	5.185	7	11.286

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2107	0.194	800	2.634
128	1126	0.364	400	2.815
256	575	0.712	200	2.875
512	330	1.241	100	3.300
1024	160	2.560	50	3.200
2048	149	2.749	25	5.960
4096	89	4.602	13	6.846
8192	98	4.180	7	14.000

Unicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 10 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3145	0.130	800	3.931
128	1612	0.254	400	4.030
256	1034	0.396	200	5.170
512	716	0.572	100	7.160
1024	421	0.973	50	8.420
2048	389	1.053	25	15.560
4096	130	3.151	13	10.000
8192	157	2.609	7	22.429

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3702	0.111	800	4.628
128	1671	0.245	400	4.178
256	894	0.458	200	4.470
512	481	0.852	100	4.810
1024	295	1.388	50	5.900
2048	130	3.151	25	5.200
4096	110	3.724	13	8.462
8192	166	2.467	7	23.714

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3401	0.120	800	4.251
128	1766	0.232	400	4.415
256	848	0.483	200	4.240
512	459	0.892	100	4.590
1024	222	1.845	50	4.440
2048	181	2.263	25	7.240
4096	201	2.038	13	15.462
8192	159	2.576	7	22.714

Unicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 5 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2332	0.176	800	2.915
128	1240	0.330	400	3.100
256	661	0.620	200	3.305
512	359	1.141	100	3.590
1024	302	1.356	50	6.040
2048	240	1.707	25	9.600
4096	200	2.048	13	15.385
8192	464	0.883	7	66.286

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2742	0.149	800	3.428
128	1657	0.247	400	4.143
256	665	0.616	200	3.325
512	359	1.141	100	3.590
1024	370	1.107	50	7.400
2048	356	1.151	25	14.240
4096	197	2.079	13	15.154
8192	502	0.816	7	71.714

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2306	0.178	800	2.882
128	1208	0.339	400	3.020
256	884	0.463	200	4.420
512	639	0.641	100	6.390
1024	466	0.879	50	9.320
2048	585	0.700	25	23.400
4096	240	1.707	13	18.462
8192	927	0.442	7	132.429

Multicast Transmission from One Transmitter to One Local Receiver
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2380	0.172	800	2.975
128	2174	0.188	400	5.435
256	857	0.478	200	4.285
512	563	0.728	100	5.630
1024	412	0.994	50	8.240
2048	659	0.622	25	26.360
4096	325	1.260	13	25.000
8192	436	0.939	7	62.286

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2123	0.193	800	2.654
128	1632	0.251	400	4.080
256	845	0.485	200	4.225
512	608	0.674	100	6.080
1024	477	0.859	50	9.540
2048	661	0.620	25	26.440
4096	354	1.157	13	27.231
8192	326	1.256	7	46.571

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2962	0.138	800	3.703
128	1821	0.225	400	4.553
256	875	0.468	200	4.375
512	581	0.705	100	5.810
1024	458	0.894	50	9.160
2048	356	1.151	25	14.240
4096	443	0.925	13	34.077
8192	326	1.256	7	46.571

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 25 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2661	0.154	800	3.326
128	1502	0.273	400	3.755
256	956	0.428	200	4.780
512	571	0.717	100	5.710
1024	478	0.857	50	9.560
2048	461	0.889	25	18.440
4096	353	1.160	13	27.154
8192	376	1.089	7	53.714

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2786	0.147	800	3.482
128	1444	0.284	400	3.610
256	849	0.482	200	4.245
512	602	0.680	100	6.020
1024	439	0.933	50	8.780
2048	382	1.072	25	15.280
4096	416	0.985	13	32.000
8192	331	1.237	7	47.286

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2674	0.153	800	3.342
128	1485	0.276	400	3.712
256	948	0.432	200	4.740
512	612	0.669	100	6.120
1024	433	0.946	50	8.660
2048	415	0.987	25	16.600
4096	369	1.110	13	28.385
8192	341	1.201	7	48.714

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 10 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2921	0.140	800	3.651
128	1404	0.292	400	3.510
256	868	0.472	200	4.340
512	644	0.636	100	6.440
1024	448	0.914	50	8.960
2048	689	0.594	25	27.560
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2563	0.160	800	3.204
128	1673	0.245	400	4.183
256	841	0.487	200	4.205
512	613	0.668	100	6.130
1024	455	0.900	50	9.100
2048	483	0.848	25	19.320
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2582	0.159	800	3.228
128	1417	0.289	400	3.542
256	841	0.487	200	4.205
512	717	0.571	100	7.170
1024	458	0.894	50	9.160
2048	3504	0.117	25	140.160
4096				
8192				

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 5 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2888	0.142	800	3.610
128	1708	0.240	400	4.270
256	1028	0.398	200	5.140
512	870	0.471	100	8.700
1024	557	0.735	50	11.140
2048	524	0.782	25	20.960
4096	432	0.948	13	33.231
8192	498	0.822	7	71.143

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3085	0.133	800	3.856
128	1747	0.234	400	4.367
256	995	0.412	200	4.975
512	880	0.465	100	8.800
1024	915	0.448	50	18.300
2048	725	0.565	25	29.000
4096	779	0.526	13	59.923
8192	653	0.627	7	93.286

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3456	0.119	800	4.320
128	1898	0.216	400	4.745
256	993	0.412	200	4.965
512	690	0.594	100	6.900
1024	532	0.770	50	10.640
2048	912	0.449	25	36.480
4096	498	0.822	13	38.308
8192	462	0.887	7	66.000

Multicast Transmission from One Transmitter to Two Local Receivers
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	1882	0.218	800	2.353
128	1092	0.375	400	2.730
256	660	0.621	200	3.300
512	481	0.852	100	4.810
1024	429	0.955	50	8.580
2048	360	1.138	25	14.400
4096	376	1.089	13	28.923
8192	329	1.245	7	47.000

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2507	0.163	800	3.134
128	1427	0.287	400	3.567
256	862	0.475	200	4.310
512	583	0.703	100	5.830
1024	381	1.075	50	7.620
2048	398	1.029	25	15.920
4096	329	1.245	13	25.308
8192	325	1.260	7	46.429

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	1831	0.224	800	2.289
128	1038	0.395	400	2.595
256	666	0.615	200	3.330
512	478	0.857	100	4.780
1024	384	1.067	50	7.680
2048	400	1.024	25	16.000
4096	328	1.249	13	25.231
8192	374	1.095	7	53.429

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 25 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2514	0.163	800	3.143
128	1313	0.312	400	3.283
256	1133	0.362	200	5.665
512	587	0.698	100	5.870
1024	452	0.906	50	9.040
2048	510	0.803	25	20.400
4096	340	1.205	13	26.154
8192	368	1.113	7	52.571

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2086	0.196	800	2.607
128	1771	0.231	400	4.428
256	1127	0.363	200	5.635
512	706	0.580	100	7.060
1024	486	0.843	50	9.720
2048	405	1.011	25	16.200
4096	330	1.241	13	25.385
8192	348	1.177	7	49.714

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3448	0.119	800	4.310
128	1341	0.305	400	3.353
256	1219	0.336	200	6.095
512	696	0.589	100	6.960
1024	480	0.853	50	9.600
2048	409	1.001	25	16.360
4096	347	1.180	13	26.692
8192	494	0.829	7	70.571

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 10 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3086	0.133	800	3.857
128	1783	0.230	400	4.457
256	1091	0.375	200	5.455
512	658	0.622	100	6.580
1024	560	0.731	50	11.200
2048	597	0.686	25	23.880
4096	548	0.747	13	42.154
8192	406	1.009	7	58.000

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2541	0.161	800	3.176
128	1729	0.237	400	4.322
256	1012	0.405	200	5.060
512	730	0.561	100	7.300
1024	636	0.644	50	12.720
2048	574	0.714	25	22.960
4096	935	0.438	13	71.923
8192	642	0.638	7	91.714

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2528	0.162	800	3.160
128	1681	0.244	400	4.202
256	1071	0.382	200	5.355
512	672	0.610	100	6.720
1024	681	0.601	50	13.620
2048	570	0.719	25	22.800
4096	393	1.042	13	30.231
8192	681	0.601	7	97.286

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 5 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2954	0.139	800	3.692
128	1790	0.229	400	4.475
256	1195	0.343	200	5.975
512	805	0.509	100	8.050
1024	625	0.655	50	12.500
2048	1264	0.324	25	50.560
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3138	0.131	800	3.922
128	1750	0.234	400	4.375
256	1210	0.339	200	6.050
512	717	0.571	100	7.170
1024	722	0.567	50	14.440
2048	764	0.536	25	30.560
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3075	0.133	800	3.844
128	1731	0.237	400	4.327
256	1125	0.364	200	5.625
512	1021	0.401	100	10.210
1024	593	0.691	50	11.860
2048	860	0.476	25	34.400
4096				
8192				

Multicast Transmission from One Transmitter to Three Local Receivers
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	1900	0.216	800	2.375
128	1103	0.371	400	2.757
256	665	0.616	200	3.325
512	525	0.780	100	5.250
1024	441	0.929	50	8.820
2048	401	1.021	25	16.040
4096	332	1.234	13	25.538
8192	324	1.264	7	46.286

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	1828	0.224	800	2.285
128	1091	0.375	400	2.728
256	690	0.594	200	3.450
512	586	0.699	100	5.860
1024	387	1.058	50	7.740
2048	359	1.141	25	14.360
4096	347	1.180	13	26.692
8192	370	1.107	7	52.857

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2637	0.155	800	3.296
128	1092	0.375	400	2.730
256	774	0.529	200	3.870
512	483	0.848	100	4.830
1024	381	1.075	50	7.620
2048	359	1.141	25	14.360
4096	333	1.230	13	25.615
8192	324	1.264	7	46.286

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 25 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2923	0.140	800	3.654
128	1305	0.314	400	3.263
256	855	0.479	200	4.275
512	759	0.540	100	7.590
1024	411	0.997	50	8.220
2048	497	0.824	25	19.880
4096	346	1.184	13	26.615
8192	394	1.040	7	56.286

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2346	0.175	800	2.933
128	1309	0.313	400	3.272
256	1173	0.349	200	5.865
512	718	0.570	100	7.180
1024	506	0.809	50	10.120
2048	455	0.900	25	18.200
4096	340	1.205	13	26.154
8192	584	0.701	7	83.429

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	2148	0.191	800	2.685
128	1244	0.329	400	3.110
256	1084	0.378	200	5.420
512	584	0.701	100	5.840
1024	452	0.906	50	9.040
2048	500	0.819	25	20.000
4096	438	0.935	13	33.692
8192	422	0.971	7	60.286

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 10 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3128	0.131	800	3.910
128	1788	0.229	400	4.470
256	1200	0.341	200	6.000
512	741	0.553	100	7.410
1024	623	0.657	50	12.460
2048	775	0.529	25	31.000
4096	456	0.898	13	35.077
8192	718	0.570	7	102.571

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3111	0.132	800	3.889
128	1737	0.236	400	4.343
256	976	0.420	200	4.880
512	677	0.605	100	6.770
1024	589	0.695	50	11.780
2048	636	0.644	25	25.440
4096	441	0.929	13	33.923
8192	851	0.481	7	121.571

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3186	0.129	800	3.982
128	1801	0.227	400	4.503
256	1030	0.398	200	5.150
512	729	0.562	100	7.290
1024	521	0.786	50	10.420
2048	436	0.939	25	17.440
4096	693	0.591	13	53.308
8192	344	1.191	7	49.143

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 5 Packets)
50K Message Size (51200 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3327	0.123	800	4.159
128	1796	0.228	400	4.490
256	1026	0.399	200	5.130
512	681	0.601	100	6.810
1024	587	0.698	50	11.740
2048	603	0.679	25	24.120
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3193	0.128	800	3.991
128	1890	0.217	400	4.725
256	1014	0.404	200	5.070
512	751	0.545	100	7.510
1024	558	0.734	50	11.160
2048				
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	3192	0.128	800	3.990
128	2009	0.204	400	5.022
256	1314	0.312	200	6.570
512	907	0.452	100	9.070
1024	567	0.722	50	11.340
2048				
4096				
8192				

Unicast Transmission from One Transmitter to One Local Receiver with No Induced Errors
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	76082	0.108	16000	4.755
128	38912	0.211	8000	4.864
256	18658	0.439	4000	4.665
512	9779	0.838	2000	4.889
1024	5487	1.493	1000	5.487
2048	3486	2.350	500	6.972
4096	2345	3.493	250	9.380
8192	2300	3.562	125	18.400

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	76715	0.107	16000	4.795
128	37733	0.217	8000	4.717
256	18918	0.433	4000	4.729
512	9594	0.854	2000	4.797
1024	5446	1.504	1000	5.446
2048	4248	1.928	500	8.496
4096	2121	3.862	250	8.484
8192	2180	3.758	125	17.440

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	74346	0.110	16000	4.647
128	37105	0.221	8000	4.638
256	18871	0.434	4000	4.718
512	9532	0.859	2000	4.766
1024	5448	1.504	1000	5.448
2048	3519	2.328	500	7.038
4096	2319	3.533	250	9.276
8192	2523	3.247	125	20.184

Unicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 25 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	51021	0.161	16000	3.189
128	28869	0.284	8000	3.609
256	14050	0.583	4000	3.513
512	7409	1.106	2000	3.704
1024	4094	2.001	1000	4.094
2048	2555	3.206	500	5.110
4096	2498	3.279	250	9.992
8192	2200	3.724	125	17.600

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	51187	0.160	16000	3.199
128	27790	0.295	8000	3.474
256	14694	0.558	4000	3.674
512	7312	1.120	2000	3.656
1024	3972	2.062	1000	3.972
2048	2600	3.151	500	5.200
4096	2629	3.116	250	10.516
8192	2170	3.775	125	17.360

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	50749	0.161	16000	3.172
128	28534	0.287	8000	3.567
256	14274	0.574	4000	3.568
512	7299	1.122	2000	3.650
1024	4071	2.012	1000	4.071
2048	2621	3.126	500	5.242
4096	1917	4.273	250	7.668
8192	2465	3.323	125	19.720

Unicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 10 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	80595	0.102	16000	5.037
128	34795	0.235	8000	4.349
256	18977	0.432	4000	4.744
512	9963	0.822	2000	4.981
1024	5618	1.458	1000	5.618
2048	5068	1.616	500	10.136
4096	4282	1.913	250	17.128
8192	4282	1.913	125	34.256

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	73550	0.111	16000	4.597
128	39202	0.209	8000	4.900
256	19526	0.420	4000	4.881
512	10426	0.786	2000	5.213
1024	5907	1.387	1000	5.907
2048	4089	2.003	500	8.178
4096	3987	2.055	250	15.948
8192	9783	0.837	125	78.264

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	74717	0.110	16000	4.670
128	36851	0.222	8000	4.606
256	18843	0.435	4000	4.711
512	10016	0.818	2000	5.008
1024	5763	1.421	1000	5.763
2048	3982	2.057	500	7.964
4096	2597	3.154	250	10.388
8192	5153	1.590	125	41.224

Unicast Transmission from One Transmitter to One Local Receiver
with Induced Errors (1 of 5 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	72083	0.114	16000	4.505
128	36554	0.224	8000	4.569
256	19927	0.411	4000	4.982
512	10476	0.782	2000	5.238
1024	6329	1.294	1000	6.329
2048	9938	0.824	500	19.876
4096	30313	0.270	250	121.252
8192	20814	0.394	125	166.512

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	80292	0.102	16000	5.018
128	37261	0.220	8000	4.658
256	19194	0.427	4000	4.798
512	10670	0.768	2000	5.335
1024	6149	1.332	1000	6.149
2048	8089	1.013	500	16.178
4096	29177	0.281	250	116.708
8192	21598	0.379	125	172.784

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	77446	0.106	16000	4.840
128	36824	0.222	8000	4.603
256	19402	0.422	4000	4.851
512	10561	0.776	2000	5.280
1024	5985	1.369	1000	5.985
2048	7247	1.130	500	14.494
4096	32681	0.251	250	130.724
8192	9644	0.849	125	77.152

Multicast Transmission from One Transmitter to One Local Receiver
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	78467	0.104	16000	4.904
128	37261	0.220	8000	4.658
256	19132	0.428	4000	4.783
512	9360	0.875	2000	4.680
1024	5554	1.475	1000	5.554
2048	4122	1.987	500	8.244
4096	2188	3.744	250	8.752
8192	1580	5.185	125	12.640

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	78622	0.104	16000	4.914
128	39548	0.207	8000	4.944
256	19767	0.414	4000	4.942
512	9332	0.878	2000	4.666
1024	5573	1.470	1000	5.573
2048	3591	2.281	500	7.182
4096	2200	3.724	250	8.800
8192	1617	5.066	125	12.936

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	78446	0.104	16000	4.903
128	37612	0.218	8000	4.702
256	20031	0409	4000	5.008
512	9127	0.898	2000	4.564
1024	5383	1.522	1000	5.383
2048	3557	2.303	500	7.114
4096	2202	3.720	250	8.808
8192	1631	5.023	125	13.048

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 25 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	68101	0.120	16000	4.256
128	26100	0.314	8000	3.263
256	13129	0.624	4000	3.282
512	6951	1.179	2000	3.475
1024	4861	1.685	1000	4.861
2048	5407	1.515	500	10.814
4096	3755	2.182	250	15.020
8192	2658	3.082	125	21.264

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	70388	0.116	16000	4.399
128	25493	0.321	8000	3.187
256	13371	0.613	4000	3.343
512	7017	1.167	2000	3.509
1024	5398	1.518	1000	5.398
2048				
4096	3841	2.133	250	15.364
8192	3789	2.162	125	30.312

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	69370	0.118	16000	4.336
128	26108	0.314	8000	3.264
256	13789	0.594	4000	3.447
512	7095	1.155	2000	3.547
1024	4704	1.741	1000	4.704
2048				
4096	3890	2.106	250	15.560
8192				

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 10 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	243436	0.034	16000	15.215
128	119628	0.068	8000	14.954
256	24038	0.341	4000	6.010
512	12599	0.650	2000	6.300
1024	10162	0.806	1000	10.162
2048	5906	1.387	500	11.812
4096	5590	1.465	250	22.360
8192	3857	2.124	125	30.856

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	247312	0.033	16000	15.457
128	118551	0.069	8000	14.819
256	29629	0.276	4000	7.407
512	12298	0.666	2000	6.149
1024	6914	1.185	1000	6.914
2048	5529	1.482	500	11.058
4096	5334	1.536	250	21.336
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	259057	0.032	16000	16.191
128	92649	0.088	8000	11.581
256	22558	0.363	4000	5.639
512	12519	0.654	2000	6.260
1024	7780	1.053	1000	7.780
2048	4982	1.644	500	9.964
4096	5142	1.593	250	20.568
8192				

Multicast Transmission from One Transmitter to One Local Receiver with Induced Errors (1 of 5 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	80825	0.101	16000	5.052
128	46106	0.178	8000	5.763
256	24424	0.335	4000	6.106
512	13842	0.592	2000	6.921
1024	7332	1.117	1000	7.332
2048	7261	1.128	500	14.522
4096	6750	1.214	250	27.000
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	74872	0.109	16000	4.679
128	37821	0.217	8000	4.728
256	20415	0.401	4000	5.104
512	12316	0.665	2000	6.158
1024	8629	0.949	1000	8.629
2048	6991	1.172	500	13.982
4096	6275	1.305	250	25.100
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	73208	0.112	16000	4.575
128	43797	0.187	8000	5.475
256	21533	0.380	4000	5.383
512	12127	0.676	2000	6.064
1024	7362	1.113	1000	7.362
2048	5979	1.370	500	11.958
4096				
8192				

Multicast Transmission from One Transmitter to Two Local Receivers
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	75383	0.109	16000	4.711
128	37114	0.221	8000	4.639
256	17976	0.456	4000	4.494
512	9495	0.863	2000	4.747
1024	5569	1.471	1000	5.569
2048	3589	2.283	500	7.178
4096	2210	3.707	250	8.840
8192	1603	5.110	125	12.824

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	78702	0.104	16000	4.919
128	40246	0.204	8000	5.031
256	17802	0.460	4000	4.450
512	9211	0.889	2000	4.606
1024	5645	1.451	1000	5.645
2048	3561	2.300	500	7.122
4096	2166	3.782	250	8.664
8192	1571	5.215	125	12.568

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	79621	0.103	16000	4.976
128	40061	0.204	8000	5.008
256	19278	0.425	4000	4.819
512	9035	0.907	2000	4.518
1024	5515	1.485	1000	5.515
2048	3561	2.300	500	7.122
4096	2249	3.643	250	8.996
8192	1544	5.306	125	12.352

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 25 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	107597	0.076	16000	6.725
128	66854	0.123	8000	8.357
256	22289	0.368	4000	5.572
512	13767	0.595	2000	6.883
1024	7111	1.152	1000	7.111
2048	4019	2.038	500	8.038
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	94507	0.087	16000	5.907
128	43887	0.187	8000	5.486
256	23579	0.347	4000	5.895
512	12887	0.636	2000	6.444
1024	8814	0.929	1000	8.814
2048	4879	1.679	500	9.758
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	78175	0.105	16000	4.886
128	37069	0.221	8000	4.634
256	18928	0.433	4000	4.732
512	10386	0.789	2000	5.193
1024	7230	1.133	1000	7.238
2048	4324	1.895	500	8.648
4096				
8192				

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 10 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	166857	0.049	16000	10.429
128	75445	0.109	8000	9.431
256	30691	0.267	4000	7.673
512	11023	0.743	2000	5.511
1024	6597	1.242	1000	6.597
2048	6218	1.317	500	12.436
4096	4872	1.681	250	19.488
8192	4418	1.854	125	35.344

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	142547	0.057	16000	8.909
128	48417	0.169	8000	6.052
256	29636	0.276	4000	7.409
512	10887	0.752	2000	5.444
1024	6601	1.241	1000	6.601
2048	6276	1.305	500	12.552
4096	3818	2.146	250	15.272
8192	4988	1.642	125	39.904

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	107963	0.076	16000	6.748
128	45863	0.179	8000	5.733
256	20945	0.391	4000	5.236
512	11812	0.694	2000	5.906
1024	6857	1.195	1000	6.857
2048	5115	1.602	500	10.230
4096	5527	1.482	250	22.108
8192				

Multicast Transmission from One Transmitter to Two Local Receivers with Induced Errors (1 of 5 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	110177	0.074	16000	6.886
128	44582	0.184	8000	5.573
256	40007	0.205	4000	10.002
512	11290	0.726	2000	5.645
1024	7324	1.119	1000	7.324
2048	6848	1.196	500	13.696
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	185618	0.044	16000	11.601
128	60400	0.136	8000	7.550
256	21411	0.383	4000	5.343
512	11971	0.684	2000	5.986
1024	12038	0.681	1000	12.038
2048	17847	0.459	500	35.694
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	209749	0.039	16000	13.109
128	58771	0.139	8000	7.346
256	22424	0.365	4000	5.606
512	11408	0.718	2000	5.704
1024	7908	1.036	1000	7.908
2048	6531	1.254	500	13.062
4096				
8192				

Multicast Transmission from One Transmitter to Three Local Receivers
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	80010	0.102	16000	5.001
128	40375	0.203	8000	5.047
256	17018	0.481	4000	4.255
512	9229	0.888	2000	4.614
1024	5577	1.469	1000	5.577
2048	3610	2.269	500	7.220
4096	2233	3.669	250	8.932
8192	1646	4.977	125	13.168

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	79856	0.103	16000	4.991
128	36210	0.226	8000	4.526
256	20407	0.401	4000	5.102
512	9342	0.877	2000	4.671
1024	5612	1.460	1000	5.612
2048	3605	2.272	500	7.210
4096	2325	3.523	250	9.300
8192	1592	5.146	125	12.736

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	74049	0.111	16000	4.628
128	40261	0.203	8000	5.033
256	18122	0.452	4000	4.530
512	9129	0.897	2000	4.564
1024	5530	1.481	1000	5.530
2048	3655	2.241	500	7.310
4096	2184	3.751	250	8.736
8192	1689	4.850	125	13.512

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 25 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	93761	0.087	16000	5.860
128	44393	0.185	8000	5.549
256	18814	0.435	4000	4.704
512	12976	0.631	2000	6.488
1024	8748	0.936	1000	8.748
2048	4014	2.041	500	8.028
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	72553	0.113	16000	4.535
128	37762	0.217	8000	4.720
256	19319	0.424	4000	4.830
512	14117	0.580	2000	7.059
1024	8890	0.921	1000	8.890
2048	4124	1.986	500	8.248
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	82543	0.099	16000	5.159
128	37336	0.219	8000	4.667
256	20611	0.397	4000	5.153
512	10317	0.794	2000	5.159
1024	5923	1.383	1000	5.923
2048	3910	2.095	500	7.820
4096				
8192				

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 10 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	123073	0.067	16000	7.692
128	48941	0.167	8000	6.118
256	29797	0.275	4000	7.449
512	10615	0.772	2000	5.308
1024	6315	1.297	1000	6.315
2048				
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	82547	0.099	16000	5.159
128	48873	0.168	8000	6.109
256	20208	0.405	4000	5.052
512	11728	0.698	2000	5.864
1024	6453	1.269	1000	6.453
2048				
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	85261	0.096	16000	5.329
128	38756	0.211	8000	4.845
256	19694	0.416	4000	4.923
512	12397	0.661	2000	6.199
1024	7684	1.066	1000	7.684
2048				
4096				
8192				

Multicast Transmission from One Transmitter to Three Local Receivers with Induced Errors (1 of 5 Packets)
1 Meg Message Size (1024000 bytes)

Run 1

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	124130	0.066	16000	7.758
128	39966	0.205	8000	4.996
256	24437	0.335	4000	6.109
512	12400	0.661	2000	6.200
1024	7978	1.027	1000	7.978
2048				
4096				
8192				

Run 2

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	94894	0.086	16000	5.931
128	39780	0.206	8000	4.973
256	26229	0.312	4000	6.557
512	10891	0.752	2000	5.446
1024	6945	1.180	1000	6.945
2048				
4096				
8192				

Run 3

Buffer Size (Bytes)	Timing (milliseconds)	Throughput (Megabits per second)	Number of calls	Latency (milliseconds per call)
64	79585	0.103	16000	4.974
128	39917	0.205	8000	4.990
256	20581	0.398	4000	5.145
512	12821	0.639	2000	6.410
1024	9192	0.891	1000	9.192
2048				
4096				
8192				

LIST OF REFERENCES

Agarwal, D.A., Moser, L.E., Melliar-Smith, P.M., and Budhia, R.K., *A reliable ordered delivery protocol for interconnected local area networks*, IEEE Comput. Soc. Press Proceedings 1995 International Conference on Network Protocols (Cat. No.95TB8122) p. 365-74, 1995.

Armstrong, S., Freier, A., and Marzullo, K., *Multicast Transport Protocol*, Network Working Group Request for Comments 1301, February, 1992. Available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Atwood, J.W., and Zhang, Y., *A definition of the XTP service and its formal specification*, IEEE Comput. Soc. Press Proceedings. 20th Conference on Local Computer Networks (Cat. No.95TB100005) p. 459-68, 1995.

Atwood, J. William, et. al., *Reliable Multicasting in the Xpress Transport Protocol*, XTP Forum, 1996.

Atwood, J. William, Catrina, Octavia, Fenton, John, and Strayer, W. Timothy, *Reliable Multicasting in the Xpress Transport Protocol*, XTP Forum, 1996.

Bauer, F., and Varma, A. *Distributed algorithms for multicast path setup in data networks*, IEEE; ACM Transactions on Networking Vol: 4 Iss: 2 p. 181-91, April 1996.

Bormann, C., et.al., *MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport*. Presented at the ICCCN '94, San Francisco, September 1994. Also available at *http:// hill.lut.ac.uk/DS-Archive/MTP.html*

Bradner and Mankin, *IPng, Internet Protocol Next Generation*, Addison-Wesley, 1996.

Braudes, R., and Zabele, S., *Requirements for Multicast Protocols*, Request for Comments 1458, Internet Engineering Task Force (IETF), May 1993. Also available at *ftp://ds.internic.net/rfc/rfc1458.txt*

Buddenberg, Rex A., Class lecture notes, 21 January 1997.

Casey, Roger, *Unicast versus Multicast Working Paper*, NRaD, January 21, 1995.

Cheeha Kim, and Jai Yong Lee, *Message ordering in multicast communication*, IEEE Comput. Soc. Press Proceedings. 1993 Global Data Networking (Cat. No.93TH0600-7) p. 186-90, 1993.

Deering, S., Estrin, D.L., Farinacci, D., Jacobson, V., Ching-Gung Liu, and Liming Wei, *The PIM architecture for wide-area multicast routing*, IEEE; ACM April 1996.

Dempsey, Bert J., and Weaver, Alfred C., *Issues in Designing Transport Layer Multicast Facilities*, University of Virginia Computer Science Report TR-90-18, July, 1990.

Electronics Technician Volume 3 – Communication Systems, Naval Education and Training Command, 1993.

Electronics Technician Volume 4 – Radar Systems, Naval Education and Training Command, 1993.

Fekete, A., *Formal models of communication services: a case study*, Computer Vol: 26 Iss: 8 p. 37-47, August 1993.

Garcia-Molina, Hector, and Spauster, Annemarie, *Ordered and Reliable Multicast Communication*, ACM Transactions on Computer Systems, Vol. 9, No. 3, August 1991, pp. 242-272.

Gopal, I, and Rom, R., *Multicasting to multiple groups over broadcast channels*, IEEE Transactions on Communications Vol: 42 Iss: 7 p. 2423-31, July 1994.

Infrastructure and Networking Research, Sandia National Laboratories, *SandiaXTP – An Object-Oriented Implementation of XTP 4.0 Derived from the Meta-Transport Library, User's Guide, Release 1.5*, 1997.

Infrastructure and Networking Research, Sandia National Laboratories, *Meta-Transport Library – A Protocol Base Class Library, Meta-Transport Library User's Guide, Release 1.5*, 1997.

Internet Multicast Backbone, World-Wide Web page updated 1996 by Bryan O'Sullivan. Available at http://www.serpentine.com/~bos/tech/mbone/

IP Multicast & MBONE. Available at http://wwwcs.cern.ch/wwwcs/public/ip/multi_mbone.html

Koifman, A., and Zabele, S., *RAMP: A Reliable Adaptive Multicast Protocol*, Submitted to INFOCOM '96, San Francisco, CA, March, 1996. Also available at http://www.tasc.com:80/simweb/papers/RAMP/abstract.htm

Levine, John R., and Young, Margaret Levine, *UNIX For Dummies, 2nd Edition*, IDG Books Worldwide, Ink, 1995.

Li Gong, and Shacham, N., *Elements of trusted multicasting*, IEEE Comput. Soc. Press Proceedings. 1994 International Conference on Network Protocols (Cat. No.94TH8002) p. 23-30, 1994.

Lin, John C., and Paul, Sanjoy, *RMTP: A Reliable Multicast Transport Protocol*, Proceedings of IEEE INFOCOM '96, March 1996, pp. 1414-1424. Also available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Macker, Joseph P., Klinker, J. Eric, and Corson, M. Scott, *Reliable Multicast Data Delivery for Military Networking*, IEEE Document 0-7803-3682-8/96, August 1996.

MBONE Website, created by Vinay Kumar, last modified on August 31, 1996. Available at: *http://www.best.com/~prince/techinfo/./*

Melliar-Smith, P.M., and Moser, L.E., *Trans: a reliable broadcast protocol*, IEE Proceedings I [Communications, Speech and Vision] Vol: 140 Iss: 6 p. 481-93, December 1993.

Montgomery, Todd, *Design, Implementation, and Verification of the Reliable Multicast Protocol*, Master's Thesis, West Virginia University, 1994.

Montgomery, Todd, electronic mail message to Petitt, David G., August 21, 1996.

MTL website, *http://www.ca.sandia.gov/xtp/mtl/*

Multicast Routing, World-Wide Web page created by Cisco Systems, Inc., posted August 3, 1995. Available at *http://www.cisco.com/warp/public/614/17.html*

Narayan, A.P., *Reliable multi-destination transfer of data in a local area network*, IEEE Comput. Soc. Press Tenth Annual International Phoenix Conference on Computers and Communications (Cat. No.91CH2959-5) p. 681-7, 1991.

Narayan, A.P., *Reliable transfer of data in a local area network with multicast distribution*, IEEE Comput. Soc Proceedings. 15th Conference on Local Computer Networks (Cat. No.90TH0335-0) p. 310-19, 1990.

Navy UHF Satellite Communication System Description, Naval Ocean Systems Center, 1984.

Petitt, David G., *Solutions for Reliable Multicasting*, Masters Thesis, Naval Postgraduate School, September, 1996.

Radioman Communications, Naval Education and Training Command, 1994.

SandiaXTP website, *http://www.ca.sandia.gov/xtp/SandiaXTP/*

Shacham, N., *Multicast routing of hierarchical data*, IEEE SUPERCOMM/ICC '92. Discovering a New World of Communications (Cat. No.92CH3132-8) p. 1217-21 vol.3, 1992.

Stallings, William, *Data and Computer Communications 4th Ed.*, Englewood Cliffs, New Jersey, Macmillan Publishing Company, 1994.

Strayer, W. Timothy, Dempsey, Bert J., and Weaver, Alfred C., *XTP: The Xpress Transfer Protocol*, Addison-Wesley Publishing, 1992.

Strayer, electronic mail message to Johnstone, George S., March 6, 1997.

Trepanier, Dennis Michael, *Internetworking: Recommendations on Network Management for K-12 Schools*, Master's Thesis, Naval Postgraduate School, September, 1995.

Tseung, L.C.N., and Yu, K., *Guaranteed, reliable, secure broadcast networks*, IEEE Comput. Soc. Press Ninth Annual International Phoenix Conference on Computers and Communications (Cat. No.90CH2799-5) p. 576-83, 1990.

Webster's II New Riverside University Dictionary, Houghton Mifflin Company, 1988.

XTP Forum, *Xpress Transport Protocol Specification, XTP Revision 4.0*, XTP Forum Inc., Santa Barbara, CA, March 1, 1995.

XTP Homepage. Available at http://www.ca.sandia.gov/xtp/xtp.html

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
   8725 John J Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Dr. W. Timothy Strayer 1
   Sandia National Laboratories
   P. O. Box 969, Mailstop 9011
   Livermore, CA 94551-0969

4. Rex Buddenberg, Code SM/Bh 3
   Naval Postgraduate School
   Monterey, CA 93943-5101

5. Dr. Don Brutzman, Code UW/Br 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

6. LT George S. Johnstone 1
   16 Mervine Street
   Monterey, CA 93940

7. LT Glenn D. Williams 1
   6-103 East Bay Village
   Middletown, RI 02842

8. Harry Gold 1
   NRaD Code 41
   San Diego, CA 92152

9. Phil Irey 1
   NSWC
   Dahlgren, VA 22448

10. William J. Atwood 1
Department of Computer Science
Concordia University
LB927-9
1455 Maisonneuve Blvd. West
Montreal, Quebec
Canada H3G 1M8

11. Greg Chesson 1
Silicon Graphics
2011 North Shoreline Blvd.
Mountain View, CA 94043

12. Alfred C. Weaver 1
Computer Science Department
Thornton Hall
University of Virginia
Charlottesville, VA 22903

13. Dave Petitt 1
526 Jefferson Drive
Lake Charles, LA 70605