PARTIAL ORDER AND PARTIAL RELIABILITY TRANSPORT SERVICE INNOVATIONS IN A MULTIMEDIA APPLICATION CONTEXT

In Two Volumes

Volume I
(Pages 1-175)

by

Phillip T. Conrad

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer and Information Sciences.

Fall 2000

PARTIAL ORDER AND PARTIAL RELIABILITY TRANSPORT SERVICE INNOVATIONS IN A MULTIMEDIA APPLICATION CONTEXT

by

Phillip T. Conrad

Approved: ____
M. Sandra Carberry, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: ____
Conrado M. Gempesaw II, Ph.D.
Vice Provost for Academic Programs and Planning

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Paul Amer, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Charles Boncelet, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Errol Lloyd, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Adarshpal Sethi, Ph.D.
Member of dissertation committee

ACKNOWLEDGMENTS

This work was supported in part by grants from the US Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement number DAAL01-96-2-0002, and by the National Science Foundation (NCR-9314056). I am grateful for this support.

I would like to express my deepest gratitude to the many people who have given their time, energy, support and prayers in support of my efforts on this dissertation. First and foremost, I would like to thank my advisor, Paul Amer, who has been an excellent mentor, guide, teacher, colleague and friend. I thank him for his professional, personal, and moral support, for his confidence in me, and his incredible patience and leadership.

Next, I would like to thank the other members of my committee: Errol Lloyd, Adarsh Sethi, and Charlie Boncelet; these folks have gone beyond the call of duty for me on many occasions, and I am deeply grateful. I would also like to thank my colleagues in the Protocol Engineering Lab at the University of Delaware, especially: Greg Burch, Tom Connolly, Armando Caro, Maruisz Fecko, Ed Golden, Sami Iren, Rahmi Marasli, Gosia Steinder and Mason Taube, for their many valuable suggestions. In particular, I would like to thank Armando, Ed, Sami and Mason for their software development assistance with UTL, ReMDoR, the Lossy Router, and the experiment scripts.

I thank my colleagues in the CIS Departments at the University of Delaware and at Temple University, for their support and confidence during the last three years of this project, when I was teaching full-time in addition to working on the research. In particular, I would like to extend my thanks to Sandee Carberry, Kathy McCoy, Bob Caviness, Frank Friedman and Giorgio Ingargiola for their guidance and mentoring during my first three years as a full-time faculty member. I would also like to thank the EECIS lab staff group at the University of Delaware, particularly Mike Davis and Ben Miller, for running the finest system administration group I have ever encountered. I am most grateful for their assistance with the many special requests our lab has made in connection with the network experimentation component of this work.

There are so many friends and family members who have offered their support and encouragement that it would be impossible to name them all in the small space alloted for these acknowledgments. However, I would like to particularly thank my mother, Greer Conrad, my father, Roddy Conrad. I would also like to thank the members of West Presbyterian Church, particularly Betty Crocker, for their prayers and gentle reminders to keep on working towards the goal. I am also very grateful for the assistance of Tom Ledbetter and Lynne Hagelin. I would also like to thank a few folks who were directly involved with helping to produce the final product. I particularly thank Sean Crist, who provided an incredible amount of help with proofreading, photocopying, collating, and generally keeping me somewhat sane through the final hours before my dissertation defense. Chris Coons, Dave Hunt, and Jeff Krehbiel also provided invaluable proofreading help at various stages.

Finally, my deepest debt of gratitude is to my partner, Bob Nieder, who has long suffered the pains of being a Ph.D. candidate spouse. His loyalty and patience are far more than I deserve, and I am so grateful for his support, love, encouragement, and faithfulness throughout the last twelve years we have been together, and especially the last eight years of dissertating. I thank him from the bottom of my heart, and dedicate this work to him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This dissertation investigates an innovation in computer communications called *Partially Ordered and Partially Reliable (PO/PR) Transport Service*. PO/PR service bridges the gap between two traditional forms of transport service: Ordered/Reliable (O/R) service, such as the Internet's Transmission Control Protocol (TCP), and Unordered/Unreliable (U/U) service, such as the Internet's User Datagram Protocol (UDP). Some applications—in particular, multimedia applications—require services that lie in between these two extremes. U/U service is insufficient for these applications, yet O/R service is too restrictive and may cause the application to pay a performance penalty.

Previous investigations of PO/PR transport service used analytic and simulation modeling to investigate the performance of an abstract PO/PR transport service called Partial Order Connection. We build on this work by first describing several innovations in PO/PR transport service developed by the dissertation author, including a new approach to coarse-grained multimedia synchronization based on extending the Object-Composition Petri Net (OCPN) of Little and Ghafoor. We then provide empirical evidence that an implementation of Partially Ordered/Reliable (PO/R) transport service called Partial Order Connection version 2 (POCv2) can provide better Quality of Service (QoS) tradeoffs to applications by providing a transport service better matched to an application's needs. In particular, we show that for multimedia document retrieval, PO/R service provides performance benefits over O/R transport service when the network loses packets.

Our experiments compare O/R service to PO/R service for a variety of documents, bit rates, window sizes, and propagation delays, with packet loss rates ranging from 0% to 30%. The results show that between 5% and 20% loss, user-perceivable improvements in progressive display are observed for bit rates between 2.4kbps to 512kbps. These results suggest that transport services providing reliable delivery over independent streams (such the emerging Internet protocol SCTP) can provide important performance benefits over lossy networks (including wireless nets or combat net radios.)

We also describe two systems developed by the dissertation author for experimenting with transport services: (1) the Universal Transport Library (UTL), a framework for implementation and testing of transport services, and (2) a Remote Multimedia Document Retrieval System (ReMDoR) that uses UTL transport services.

Chapter 1

INTRODUCTION

1.1 Problem statement

This dissertation investigates an innovation in computer communications called *Partially Ordered and Partially Reliable (PO/PR) Transport Service*. This innovative technique includes several features that are designed to provide computer applications with more flexibility in the way they use a packet-switched communications network such as the Internet. In particular, PO/PR transport service bridges the gap between two traditional forms of transport service:

- Ordered/Reliable (O/R) service, such as the Internet's Transmission Control Protocol (TCP), and
- Unordered/Unreliable (U/U) service, such as the Internet's User Datagram Protocol (UDP)

The premise of this work is that some applications—in particular, multimedia applications—require services that lie in between these two extremes. Unordered/Unreliable service is insufficient for these applications, yet Ordered/Reliable service is too restrictive and may cause the application to pay a performance penalty. The goal is to determine whether better Quality of Service (QoS) tradeoffs and/or performance improvements can be obtained by using a transport service in between these two extremes—one that is better matched to an application’s needs.

Previous investigations of PO/PR transport service used analytic and simulation modeling to investigate the performance of an abstract PO/PR transport service called Partial Order Connection (POC) (Marasli, 1997). By contrast, the **problem statement** for this dissertation is:

> ***To determine through experimentation with real systems the extent to which PO/PR transport service can provide performance benefits for real applications***.

Towards this end, the author designed a PO/PR transport service called Partial Order Connection, version 2 (POCv2). POCv2 is a second version of the abstract PO/PR transport service (POC), first proposed in (Amer et al., 1994), and then investigated by Marasli through analysis and simulation (Marasli et al., 1996, 1997a, 1997b). POC provides a good basis for reasoning about partial order protocols and doing simulation and analysis. However, as we shall show later in this chapter, POC lacks certain features necessary for experimentation and/or deployment with real applications.

To perform performance experiments comparing POCv2 and other PO/PR transport services to traditional transport services, the author designed:

- a framework for implementation and testing of experimental transport services (including POCv2 and others),
- an application called ReMDoR that benefits from a PO/PR service, and
- a framework for repeating performance experiments with ReMDoR under controlled conditions.

The author then supervised and participated in the development of these systems, and then used them to carry out performance experiments comparing Partially

Ordered/Reliable (PO/R) service (a subclass of PO/PR service) to ordered/reliable (O/R) and unordered/reliable (U/R) service.[1] The remainder of this dissertation describes the systems that were developed to conduct these experiments and the results obtained. The next section outlines our key results.

1.2 Key results of this dissertation

In keeping with the problem statement above, our *central* result is the analysis of *performance experiments illustrating a range of parameters where PO/R transport service can provide better performance than O/R service for a multimedia document retrieval system*. This result is important for at least two reasons. First, prior to the publication of this data, all claims about the benefits of PO/PR service have been based on either intuitive arguments, as in (Amer et al., 1994), or simulation or analysis of abstract applications, as in (Marasli et al., 1997a, Marasli 1997b). Putting this data in a real application context provides an important grounding for delineating the practical benefits available from PO transport service. Second, the fact that there is benefit from a PO/R service over an O/R service provides important motivation for future work to extend our empirical study to an investigation of the benefits of PO/PR service.

The experiments (described in detail in Chapter 5) compare O/R service to PO/R service (and in one case, also to U/R service) for a variety of documents, bit rates, window sizes, propagation delays. The experiments were carried out at loss rates ranging from 0% to 30%, with the emphasis on loss rates between 5% and 20%. We defer the presentation of specific numerical results until Chapter 5; nevertheless,

---

[1] We defer experimentation with PO/PR service to future work; this is consistent with the approach taken in a previous dissertation in this area (Marasli 1997b),

we can preview for the reader some of the general trends that were observed. The two most important trends are ones that confirm the value of PO/R service:

(1) At 0% loss, there is little to no benefit or penalty for using a PO/R service rather than an O/R service, regardless of the values of any other parameters.

(2) At loss rates higher than 0%, PO/R service provides a clear performance advantage over O/R service under some conditions, for some documents.

In particular, we note the following about the conditions under which PO/R service outperforms O/R service, and the nature of the advantages:

(3) In general, PO/R service provides faster progressive display of information than O/R service when the loss rate is larger than 0%. As the loss rate increases, the advantage of PO/R service steadily increases. Thus PO/R service can make an application more robust to a certain amount of packet loss, up to a certain point. As the loss rate increases, after a certain loss rate is reached, the gain may be moot since both PO/R and O/R services are unacceptably bad.

(4) The size of the gain due to partial order is sensitive to changes in the flow control and/or congestion control schemes, including the sender and/or receiver window sizes, and whether or not TCP-friendly mechanisms such as slow start and congestion avoidance are used. However, we can show gains *both* when TCP-friendly mechanisms are used, *and* when they are not.

(5) PO/R service provides benefits for several different kinds of documents, including: small documents with no temporal dimension (similar to web pages with multiple GIF or JPEG images), simple multimedia documents with images and audio in parallel, and larger documents with complex synchronization relationships among multiple data streams.

In addition to this central result, we also present several other key results:

- We describe two *experimental systems* that were designed and developed as part of this dissertation work:

- a Universal Transport Library (UTL) providing a framework for development and testing of experimental transport services, and
- a Remote Multimedia Document Retrieval System (ReMDoR) that can operate over multiple transport services.

While these systems were originally developed for the author's dissertation work, they proved useful for other research as well—particularly research into *network-conscious image compression* (explained further in Section 1.6.3). In addition to this author's four ReMDoR/UTL related publications, five other authors have published a total of eight other journal articles, conference papers, and/or PhD, MS or BS honors theses that either: (1) included experimental results derived using ReMDoR and/or UTL, and/or (2) described design, architecture and implementation issues related to the development of ReMDoR and/or UTL.

- We describe several *innovations in PO/PR transport service* developed by the dissertation author, thus extending the previous work on PO/PR service. Some of the innovations described here were actually implemented as part of this dissertation, as they were necessary for the completion of the performance experiments. Others have not been implemented to date; for these we provide intuitive arguments why they should provide performance benefits. The latter set of innovations provides a number of opportunities for future study (see Chapter 7).

1.2 Structure of the dissertation

The dissertation is divided into seven chapters, as shown in Figure 1.1. Chapter 1 provides an introduction to the dissertation, including the problem statement, an overview of the motivation for studying partially ordered and partially reliable transport services, and an overview of the motivation for studying the performance of these services in the context of multimedia applications. Chapter 2 provides a more in-depth look at some of the innovations in PO/PR transport service proposed in this dissertation. Chapters 3 and 4, respectively, describe the design and implementation of the Universal Transport Library (UTL), a framework for the development and testing of experimental transport services, and the Remote Multimedia Document Retrieval System (ReMDoR) that can operate over multiple transport services.

Chapter 5 presents the core of this dissertation: performance results from experiments designed to evaluate PO/PR service. This chapter includes an overview of the experimental methods used, and results from experiments designed to validate the experimental framework. This is followed by a complete discussion of experiments comparing ordered/reliable (O/R) service, partially ordered/reliable (PO/R) and unordered/reliable (U/R) service. We present results obtained by measuring a specific application: remote multimedia document retrieval via ReMDoR.

In Chapter 6 we turn to a different question: the communications and processing overhead of PO/PR service. Two algorithmic questions are considered: how to encode a partial order for effective transmission and processing, and how to design efficient algorithms for the extra processing that sending and receiving transport entities must perform to provide PO/PR service. Finally, Chapter 7 provides a summary of the key results from this dissertation, and suggestions for future work.

The appendix describes two sample multimedia documents (named “`paris.pmsl`” and “`military.pmsl`”) that are used throughout this dissertation as examples to illustrate ideas, and as the basis of the performance experiments.

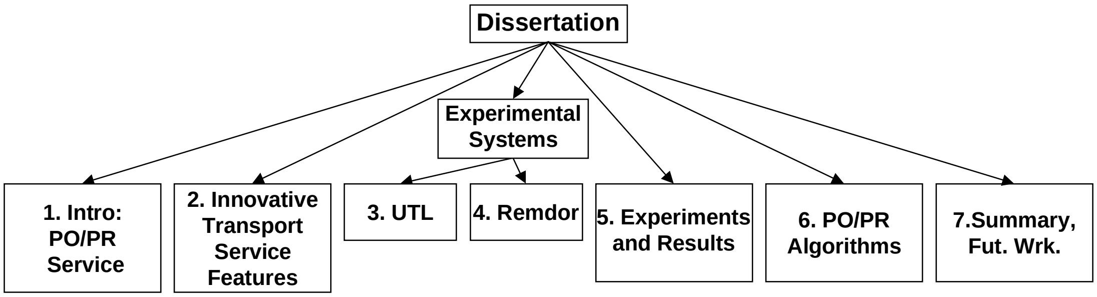

Figure 1.1 Structure of the dissertation

1.3 Background

It is assumed that the reader is familiar with concepts of computer networking, and with the transport layer in particular. For example, the reader is assumed to be familiar with:

- the OSI model reference model; in particular, the transport layer of that model
- terms such as Protocol Data Unit (PDU) and Service Data Unit (SDU)
- the concepts of service, protocol, implementation and peer entities
- basic concepts of the TCP/IP protocol suite, such as the role of TCP, UDP and IP
- the concept of Quality of Service (QoS)
- what an Internet *Request for Comments (RFC)* document is

The dissertation author is a co-author of (Iren et al., 1999a) which provides a tutorial and survey on the transport layer; the tutorial section in particular provides a good summary for readers not familiar with the terminology above.

1.4 Partial order and partial reliability transport service

A distinction can be made between *qualitative* QoS parameters and *quantitative* QoS parameters. Examples of qualitative QoS parameters include:

- **order**: Is the sending order preserved; that is, are messages received out-of-order resequenced before delivery?
- **reliability**: Are lost messages recovered through re-transmission or forward error correction?
- **duplication**: It is possible for a message to be delivered more than once?
- **flow-control**: Are mechanisms in place to prevent a fast sender from overwhelming a slow receiver, resulting in either data loss or waste of network resources?

Examples of quantitative QoS parameters include:

- **delay**: How long does it take for messages to travel from sender to receiver?
- **throughput**: How many messages are delivered per unit of time?
- **jitter (or burstiness)**: Do messages arrive in a predictable steady flow, or in huge bursts with long quiet periods in between? More formally, what is the variance of packet interarrival times?

One fundamental transport layer design problem is making appropriate tradeoffs between qualitative and quantitative QoS parameters. Choosing an appropriate tradeoff is important, because while transport services are often called upon to provide a QoS that is an enhancement of the underlying network, improving the performance as measured by one QoS parameter usually involves degrading the performance of another QoS parameter. For example, TCP provides a reliable service on top of the unreliable IP network protocol by means of retransmissions, but does so at the expense of introducing additional end-to-end delay. The selection of which transport mechanisms are appropriate for a given application is often a matter of considerable debate. For example, while the prevailing view is that retransmission is inappropriate for multimedia data because of its real-time nature, (Little and Ghafoor, 1991), some authors describe circumstances in which retransmission is appropriate (Dempsey et al., 1996.)

Commonly, transport protocol selection is a choice between extremes. For example, in the Internet protocol suite we note that the service provided by TCP (RFC793) is ordered, reliable, no-duplicates and flow-controlled, while the service provided by the *User Datagram Protocol* (UDP, RFC768) is unordered, unreliable, may duplicate messages, and is not flow-controlled. The fact that these protocols provide service at the extremes of each of these four QoS axes creates a dilemma for the designer of an application whose needs reside in the middle. When designing an application that will run over Internet protocols, today's implementer typically has three choices: (1) use TCP, (2) use UDP as is, or (3) implement a custom transport protocol on top of UDP (a considerable software development effort.)

As explained below, choosing either TCP or UDP when neither is appropriate has negative consequences. Thus, many applications utilize the third choice: building the needed transport functionalities from scratch, as an application specific transport protocol layered on top of UDP, for example, see (Jacobs and Eleftheriadis, 1997; RealNetworks, 1997.) Implementing a custom protocol allows the application designer to choose exactly what features to implement based on the requirements of the application. However, implementing transport protocol features at the user-level of the operating system is non-trivial. Two issues are particularly difficult: (1) managing the context switching between asynchronous protocol events, such as timer expiration and packet arrival, and the rest of the application code, and (2) getting flow-control/congestion-avoidance to operate correctly. In the latter case, to truly test the correctness of the design and implementation requires simulation and/or implementation on a wide scale. The complexity of designing and testing the operation of retransmission timers, resequencing of data, buffers, round-trip-delay estimation, and flow-control/congestion avoidance may exceed the complexity of the application itself!

We argue that given a reasonable alternative, application designers would prefer to avoid programming transport layer functionality. Therefore we present an alternative: a standardized transport service providing the flexibility to specify reliability, ordering, flow-control, and duplication at a finer granularity than either TCP or UDP. This flexibility allows application designers to focus their efforts on their application rather than on transport layer details.

1.4.1 Problems with using TCP or UDP

In a protocol environment where only the extremes of transport service are provided, some applications cannot find a perfect home. Consider the retrieval of objects that are part of a multimedia presentation. For some objects, no loss is permissible (e.g., text, some still images, control information) while for other objects some loss may be permissible (e.g., audio and video streams, images that are merely decorative). Also, the order of presentation of objects may be defined by a partial order rather than a total order, as is the case for document synchronization requirements described by an Object Composition Petri Net (OCPN) (Little and Ghafoor, 1991.) We describe the service required by such an application as *partially-ordered/partially-reliable*. Partially-reliable refers to the notion that some objects must be delivered reliably, while others may be lost if necessary. Partially-ordered refers to the fact that data sequencing requirements are expressed as a partial order rather than as a linear order.

For such an application, TCP provides more reliability and resequencing than is necessary at the expense of extra delay and reduced throughput. Extra delay may result in annoying discontinuities in the playback of continuous media such as audio or video data. However, TCP has the advantage of providing flow control and congestion control algorithms that have been tested for nearly two decades, and scale well to a global internet.

UDP, on the other hand, provides a “best effort” service with no guarantees whatsoever. On a lightly loaded LAN where the underlying network is inherently reliable (at least, for most practical purposes, i.e., packet loss probabilities of 10^{-9} or less), a best-effort service may be perfectly acceptable. Over longer Internet distances, where packet loss probabilities routinely range anywhere from negligible to over 50%, UDP may be fine one day and completely unacceptable the next. Our anecdotal experience with the MBONE tools for Internet video-conferencing (i.e., nv, vat) suggests that sustained packet drop rates between 7-15% are common, and that drop rates as high as 50% do occasionally occur. There are several studies that support our anecdotal experience. (Diot and Gagnon, 1999) cite similar experiences with packet loss in the Internet. (Bolot, 1993) observed packet loss rates around 10% on connections between INRIA, Sophia-Antipolis, France, and the Univ. of Maryland, College Park MD, USA. While (Paxson, 1996) does not directly measure packet losses, the high frequency of routing anomalies he cites lends support to the notion that the Internet provides a highly variable quality of service with a significant rate of failure.

Some argue that since loss in the Internet is commonly due to buffer overflows, bandwidth reservation and improved congestion control methods are needed, and if implemented, will eliminate packet loss as a significant problem. RSVP (Zhang et al.,1993) and YESSIR (Pan and Schulzrinne, 1998) are two examples of such reservations schemes. Our sense is that in spite of excellent research efforts related to guaranteed QoS through reservations, there will always be environments where reservation mechanisms are infeasible to implement, or fail to provide the necessary QoS (e.g., a disaster situation or battlefield scenario involving intermittent jamming). In these cases the loss rate of the underlying network may be higher than an application's tolerance for loss. Furthermore, because UDP is not flow-controlled, unless the application implements its own flow-control mechanism, an application using UDP may flood the network and/or the receiver with packets at a rate faster than either can handle, thus creating another source of packet loss.

1.4.2 Partially-ordered/partially-reliable transport service as an alternative

What is desirable is a standardized transport service, or a library of functions, modules, or objects, that applications can utilize to gain flexible control over the ordering and reliability of individual objects. Such a service or library would allow applications to achieve an appropriate balance among various QoS parameters without having to “reinvent the transport-layer wheel” with every application. Such an approach is consistent with Application Level Framing (ALF) as proposed in (Clark and Tennenhouse, 1990). This dissertation presents two technologies to address the need for additional flexibility at the transport layer. The first is the POCv2 protocol, which provides a *partially-ordered/partially-reliable* (PO/PR) transport service. Chapter 2 includes an overview of partially-order/partially reliable transport services, including a summary of previous and related work. The second is a more general mechanism: the *Universal Transport Library* (UTL), which provides a framework for the development of transport services that provide flexible QoS tradeoffs. UTL is described in more detail in Section 1.7.1, and Chapter 3.

In addition to allowing an application to request the precise level of reliability and ordering required, POCv2 and the UTL provide an additional benefit that neither TCP nor UDP provides: a mechanism that facilitates coarse-grained synchronization of multimedia objects. This synchronization feature is described in more detail in Chapter 2.

1.5 PO/PR Transport service and multimedia document retrieval

Previous work on partially-ordered service has produced quantitative results for delay and throughput gains for PO/PR transport service. (Marasli et al. 1996, Marasli et al., 1997a, Marasli 1997b). However, these quantitative results were derived for an abstract PO/PR service, and not directly related to any concrete application. A goal of this dissertation is to put such quantitative measures into an application context so that such results can be interpreted in terms of their impact on an end user. The application chosen for this purpose was remote multimedia document retrieval; that is, retrieving a multimedia document from a server on the Internet, and presenting this document as it is retrieved from the server. The remainder of this section explains why this application is particularly suited to a study of PO/PR services. It also introduces two key themes of our work: (1) the benefits of providing graceful degradation of multimedia documents, and (2) the benefits of integrating the transport order and reliability features with the coarse grained synchronization mechanism of the multimedia application.

1.5.1 Graceful degradation of multimedia documents

Many systems now exist that allow authors to construct pre-orchestrated multimedia documents. One of the most popular commercial systems is Macromedia Director. The proceedings of the IEEE and ACM Multimedia conferences contain examples of research systems; a survey of such systems appears in Chapter 4.

Multimedia documents consist of objects such as still images, text, audio clips and video clips, which are pre-arranged according to a temporal scenario. Various schemes exist for expressing temporal scenarios (Pérez-Luque and Little, 1995). During the playback of such a document, object presentation proceeds according to this temporal scenario until some event occurs which stops or resets it--for example, a user interaction point is reached, or a user presses a pause button. Typically, a multimedia workstation with sufficient CPU, memory, and I/O capabilities can present a document in compliance with its temporal scenario, provided that the channel delivering the information is ordered and error-free.

However, suppose the document is stored on a remote file server, and the channel delivering this information is the Internet. In this case, network errors and delays may wreak havoc with attempts to present the document correctly.

We propose that in such situations, it is appropriate to provide for *graceful degradation* of the multimedia document presentation. Graceful degradation is helpful in multimedia documents where not all objects have equal importance, or the same quality-of-service requirements; that is, some objects are essential to document content, while others are nice to have, but optional.

Graceful degradation is also helpful when some objects must be presented in a specific order, while other objects can be presented in an order different from their transmission order, with no loss of quality. For example, in a document describing a simple repair to a piece of equipment, “step 1” should be presented before “step 2”. Now, suppose the same document also contains three images that should be presented roughly simultaneously. If two of them show up, and one has to be retransmitted, in many cases it is desirable to go ahead and present the images that arrived while waiting for the retransmission of the missing image.

1.5.2 Traditional transport protocols are not satisfactory

Given that we want to provide for graceful degradation, what transport protocol should be used for multimedia objects? We argue that classic transport services such as TCP and UDP are ill suited to this application, and investigate PO/PR service as an alternative. The ReMDoR system developed for this dissertation provides users with the capability to author multimedia documents and place them on a server for remote retrieval and display via the Internet. The purposes of the ReMDoR system are:

- to show how a PO/PR transport service facilitates coarse-grained synchronization of multimedia objects and graceful degradation during times of network stress,
- to demonstrate the mechanisms needed to implement a PO/PR transport protocol in practice, and
- to demonstrate and quantify performance improvements when a PO/PR transport service is used instead of an ordered/reliable service (e.g., TCP) or an unordered/unreliable service (e.g., UDP).

1.5.3 Traditional authoring systems are not satisfactory.

Note that the manner in which a document should be gracefully degraded is largely dependent on the intentions of the document author; it cannot be deduced solely from structural aspects of the document such as the type of objects (graphics, sound, text, etc.) For example, consider a sound clip. In one document, a sound clip may be only a sound effect to draw attention to a certain visual image on the screen. If this image is also highlighted in other ways (e.g., with color), then the sound effect may be helpful, but non-essential. It would be acceptable, for example, to allow for small gaps in the sound object (say, gaps of up to 20 ms); these might create small cracks or pops in the sound, but the sound would still serve its intended purpose. On the other hand, if the document were intended as a tutorial for peacekeeping troops to teach useful phrases in the local language, a sound clip might be the most important content in the document. In this case, even a small amount of infidelity in the sound could render the document useless for its intended purpose.

Current commercial authoring systems (e.g., Macromedia Director and Authorware) generally assume either

- a perfect, high-bandwidth channel (e.g., from a local CD-ROM or DVD device), or
- transmission over the Internet (i.e., the Web) using a reliable ordered channel (that is, TCP), with a medium-to-high bitrate (i.e., 33.6–56kbps to 1.544 Mbps or higher).

In general, current systems do not provide the capability to author for graceful degradation; that is, authors cannot specify the relative importance of objects in a document, or specify multiple orders for object presentation.

Some systems for Web authoring do provide the capability to make multiple versions of a document available: for example, a graphics intensive version for those connecting via a high-speed connection to the Internet, and a text-only version for those with a slow connection. There are new features that make this easier in the newest version (v1.1) of the Hypertext Transfer Protocol (HTTP) used for retrieving Web documents. In these systems, the user (or the user's browser) must choose *in advance* to retrieve a lower-quality version of the document content.

It is certainly useful for a user or a browser to be able to convey to the server that the user already *knows* network conditions are less than ideal, and that the user will accept a lower quality version of the document. However, providing multiple documents for users to choose under different network conditions is fundamentally different from the kind of graceful degradation we address. We address the case where network conditions are either unknown a priori, or suddenly become worse in the middle of a transmission. Our vision of graceful degradation is that when performance is (perhaps) unexpectedly bad, the application and the transport protocol should have already marked the most important information and sequence constraints for preservation. The application and transport entities can then immediately take steps to preserve the most important document elements and relationships, even while discarding or disregarding others. It is this kind of graceful degradation that we explore in the ReMDoR system as we consider how PO/PR transport may be of benefit.

1.6 Overview of systems developed for this dissertation

1.6.1 Universal Transport Library (UTL)

The central goal of this dissertation is to investigate whether PO/PR transport service provides a measurable benefit for some application. To show a measurable benefit, we need an application that can be demonstrated over more than one transport service. The ReMDoR application described in Chapter 4 was developed for exactly this purpose.

However, early in the development of ReMDoR, we recognized that designing an application that can run over multiple transport protocols presents certain difficulties (as we explain in Section 3.1.) To overcome these difficulties, we developed the Universal Transport Library (UTL). UTL is a library of transport layer software that can be linked in with an application, to provide a range of transport services through a single API. The transport services provided in UTL include simple wrappers for TCP and UDP, as well as a range of PO/PR transport services. The transport layer functionality in UTL is implemented at user-level rather than in the kernel, and sits in between the application and the regular UDP and TCP services provided by the operating system.

UTL provides benefits both for developers of new transport layer services, and developers of applications that want to take advantage of various kinds of transport layer service. For developers of transport layer services and protocols, UTL provides a framework for rapid prototyping of transport layer implementations. For application writers, UTL provides a library of various transport services that can be accessed through a single API.

1.6.2 Overview of the Remote Multimedia Document Retrieval system (ReMDoR)

ReMDoR is a multimedia document retrieval system that allows authors to specify synchronization requirements and varying degrees of reliability for multimedia elements (Conrad et al., 1996, Conrad et al., 1998). The key motivation for ReMDoR was the investigation of partial order and partial reliability transport protocols in the context of multimedia document retrieval. Because it was infeasible to incorporate partial order and partial reliability into existing multimedia document retrieval systems, it was necessary to develop a simple multimedia document retrieval system in order to carry out this research.

The basic model is similar to that of the World Wide Web; documents are available on a server and are retrieved via a browser. The ReMDoR browser has a look and feel that is similar to traditional web browsers, making experimentation convenient, and helping to demonstrate the idea of multimedia document retrieval in a familiar context. However, unlike Web documents, ReMDoR documents are *temporal*—they have a time dimension requiring synchronization of multimedia elements. ReMDoR has capabilities that support experimentation with innovative protocols and data compression techniques, such as:

- the ability to select from a wide range of transport services and transport service features (via UTL),
- the ability to record statistics about performance on an object-by-object basis
- features to support the automation of repeated performance experiments, and
- the ability to easily incorporate new image formats, such as the formats required for research into network-conscious image compression: Network Conscious GIF (Amer et al., 1999)., the SPIHT Wavelet format (Said and Pearlman, 1996; Iren, 1999):), and the Network Conscious Wavelet format. (Iren et al., 1998; Iren and Amer, 2000).

1.6.3 Publications based on these tools

UTL and ReMDoR have proven useful in research beyond this dissertation, chiefly, in the NETCICATS project (Iren et al., 1998a,1998b,1998c,1999b; Amer et al. 1999). In addition, the implementation work has provided opportunities for one MS Thesis, and one undergraduate Honors thesis. Table 1.1 provides a summary of the various publications, theses, and dissertations based on research that uses UTL and ReMDoR. The remainder of this section summarizes highlights of this related work.

Table 1.1 Publications related to UTL and ReMDoR implementation

1st Author	Publications	Citations
Caro	Undergraduate Honors Thesis	CIS Dept., U. Del. (Caro, 1998)
Conrad	Two Conference Papers Two Workshop Papers	MMCN’96 (Conrad et al., 1996) MILCOM’98 (Conrad et al., 1998) IWQoS’97 (Conrad et al., 1997) Sync’95 (Conrad et al. 1995)
Golden	MS Thesis	CIS Dept., U. Del (Golden, 1997)

The *Network Conscious Image Compression and Transmission System (NETCICATS)* project, which is the Ph.D. dissertation work of Sami Iren (Iren, 1999b) uses UTL and ReMDoR as the central tools for conducting performance experiments. *Network-conscious image compression* is an approach to image compression that seeks not solely to maximize compression, but rather to optimize *overall* performance when compressed images are transmitted over a lossy packet-switched network such as a battlefield network. Using an Application Level Framing philosophy, an image is compressed into path-MTU sized Application Data Units (ADUs) at the application layer.

Performance experiments in NETCICATS typically involve comparison of a traditional image compression scheme requiring a more restrictive transport protocol (e.g., Ordered/Reliable service) against an image compression scheme that may require more bits per pixel, but which allows a less restrictive transport protocol. The key idea is to find the loss rates at which the best trade-off is made; that is, the loss rates above which the ***penalty*** of more bits per pixel is outweighed by:

- the ***benefit*** of being able to deliver information out-of-order (thus providing better progressive display), ***and/or***
- the ***benefit*** of being able to tolerate a certain degree of loss while still maintaining acceptable image quality.

Thus, for evaluations of NETCICATS, three things are essential:

(1) The ability to easily put a single application on top of various transport services (an ability provided by UTL)

(2) The ability to retrieve and display different image formats over various transport services, and add new image formats easily (an ability provided by ReMDoR.)

(3) Facilities for the automation of repeated performance experiments, and the gathering of performance statistics (also provided by ReMDoR).

Thus, in addition to providing a platform for the evaluation of PO/R transport service vs. O/R transport service, ReMDoR and UTL have been critical to the network-conscious image research of Iren.

UTL and ReMDoR have also provided an opportunity for significant participation by MS and undergraduate students in research, and holds the promise of continuing to do so for future students of the dissertation author. For his MS thesis, (Golden 1997), described the first implementation of the KXP mechanism of UTL (see Sections 3.3.7, 3.6), and introduced a new transport protocol, the Timed Reliability

Unordered Message Protocol (TRUMP). In this protocol, messages can be given timestamps after which their reliability expires. TRUMP is particularly appropriate for military applications such as the Fact Exchange Protocol, where the value of a particular message becomes less valuable with time (Chamberlain, 1994). UTL provides a suitable environment for experimenting with this technique; implementation of TRUMP is among the projects suggested for future work. As an undergraduate honors thesis, (Caro, 1998) did most of the programming work for the second major version of ReMDoR, including the addition of several key features we describe in Chapter 4; he is currently working on a Java interface for UTL. The dissertation author plans to involve students in several UTL and ReMDoR projects following the completion of this dissertation; including porting UTL and ReMDoR to Linux, and building a Java version of ReMDoR over Caro's Java interface to UTL.

1.7 Overview of performance experiments

The performance experiments presented in this dissertation are divided into two major groups, labeled N for NETCICATS, and R for ReMDoR. The NETCICATS group of experiments compares O/R protocols with PO/R and U/R protocols. NETCICATS is built on top of ReMDoR, but concentrates on fixed images with no temporal dimension, and thus uses only a subset of ReMDoR's full functionality. The ReMDoR group of experiments focuses on the difference in performance between O/R and PO/R service. The ReMDoR group incorporates experiments with temporal documents that cover a range of complexity in terms of multimedia content and synchronization requirements. Chapter 5 provides details concerning these experiments. All use the basic setup illustrated in Figure 1.2. This architecture consists of a server and client, with an unreliable network in between.

Two experimental tools developed for this dissertation provide control over the properties of this network: a lossy router for simulating packet loss, and a packet reflector for simulating various bitrates and propagation delays. Section 5.3 provides more details concerning these tools.

1.8 Chapter summary

We have presented the problem statement for this dissertation, which is to determine through experimentation with real systems the extent to which partially-ordered/partially-reliable (PO/PR) transport service can provide performance benefits for real applications. The motivation for PO/PR service is to provide applications with more flexibility in making tradeoffs between QoS parameters such as order and reliability, and quantitative QoS parameters such as throughput, delay, and buffer utilization. Previous work in this area used analytic modeling and simulation to predict the performance benefits of PO/PR service, while this work focuses on putting these benefits into a concrete application context so that the impact on the end user can be assessed. Multimedia document retrieval is used as an example application that can benefit from PO/PR service.

We have described two experimental systems designed to conduct experiments comparing the performance of a remote multimedia document retrieval system over PO/PR transport service: (1) UTL and (2) ReMDoR. We provided an overview of the design and results of performance experiments conducted for this dissertation using these tools. We also noted that while the author designed UTL and ReMDoR for his own purposes, they have proven useful beyond the scope of the author’s work.

The next chapter (Chapter 2) focuses on ***transport protocol ideas***. This is followed by two chapters (Chapters 3 and 4) focusing on ***experimental systems***, covering UTL and ReMDoR, respectively. This is followed by a chapter on ***performance results*** (Chapter 5), a chapter on ***algorithms*** for PO/PR protocols (Chapter 6), and a ***summary and description*** of future work (Chapter 7).

Table 1.2 Performance Experiment Groups

Grp	**Explanation**	**Evaluation of**	**number of experiments**	**described in sections**
N1	NETCICATS as motivation for unordered service, limiting case for gains from partial order	U/R vs. PO/R vs. O/R	1	5.2
R1	ReMDoR application, 8 parallel images, slow PPP link, combat net radio speeds.	PO/R vs. O/R	4	5.3
R2	ReMDoR application, 8 parallel images, Narrowband ISDN speed (128kbps)	PO/R vs. O/R	4	5.4
R3	ReMDoR application, 8 parallel images, Narrowband ISDN speed (128kbps), effect of various bitrates	PO/R vs. O/R	1	5.5
R4	ReMDoR application, audio in parallel with images.	PO/R vs. O/R	4	5.6
R5	Excerpt from a complete document with audio, parallel images streams, complex synchronization relationships	PO/R vs. O/R	2	5.7
Total			16	

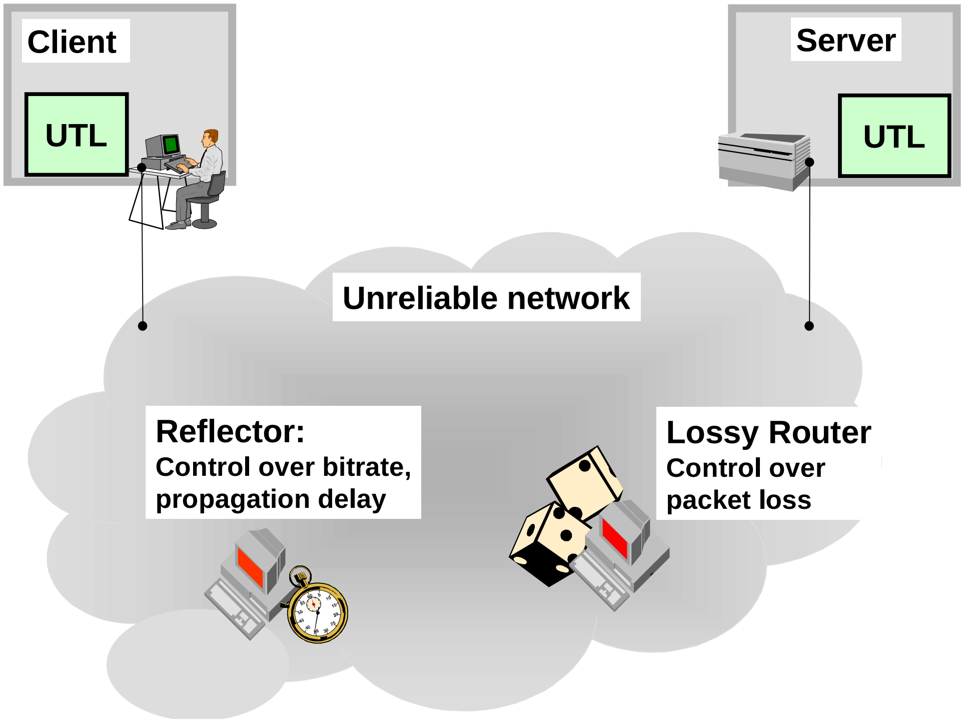

Figure 1.2 Experimental setup for performance experiments