Graceful Degradation of Multimedia Documents via Partial Order and Partial Reliability Transport Protocols [*]

Phillip T. Conrad
Paul D. Amer
Rahmi Marasli

Computer and Information Science Department
University of Delaware
Newark, DE 19716 USA
Email: {pconrad,amer,marasli}@cis.udel.edu

1 Introduction

The goals of our research on *partial-order/partial-reliability* (PO/PR) transport protocols [1] are to (1) determine whether there exist applications for which such protocols can provide better service than currently available protocols, (2) determine how to achieve this improvement in service (i.e., how to implement and use such protocols) and (3) describe and quantify this improvement in service.

Performance study of (PO/PR) transport protocols has progressed from abstract modeling and simulation [3] to a stage where a concrete target application must be chosen. This target application will be used as a reference for designing and interpreting future simulation work. In addition, a working prototype will be implemented for the purpose of measuring empirical performance results.

The target application that has been chosen is multimedia document retrieval. Because of this, synchronization will play a major role in the design and implementation of both the document specification model and the working prototype. It will also play a role in the evaluation of the results. Two key questions are: (1) Can a PO/PR service provide better performance and/or better "graceful degradation" than current protocols for a multimedia document retrieval system built on a network that loses and reorders packets? (2) Is it any easier to implement synchronization techniques when the underlying transport protocol provides PO/PR service?

This paper describes work that is just getting underway to explore these and other questions. Section 2 provides an overview of POC and PO/PR protocols. Section 3 motivates "graceful degradation" of multimedia documents. Section 4 presents an example illustrating how a PO/PR protocol can be used to facilitate graceful degradation. Section 5 reviews some of the synchronization issues we expect to encounter in the development of our prototype system.

2 Description of *Partial Order Connection* (POC)

Three fundamental *quality of service (QoS)* parameters for a transport-layer service are *loss*, *order*, and *duplication*. Some authors lump these together under the term "reliability," using the term "reliable" to refer to a message delivery service where no messages are lost, where all messages are delivered in the same order as submitted, and where no duplicates of messages are delivered.

[*] This work supported, in part, by the National Science Foundation (NCR-9314056) and by the US Army Research Office (ARO) (DAAL03-92-G-0070, DAAH04-94-G-0093).

To classify transport QoS with greater precision, we have restricted the term "reliability" to refer only to whether messages may be lost, defining a *reliable* service as one that allows no loss whatsoever. In this way, reliability becomes orthogonal to the service features of order and duplication. The term "order" refers to whether or not the order in which objects are sent is preserved in delivery. An *ordered* service delivers objects in the same total order as they were presented to that service. "Duplication" refers to whether or not multiple copies of the same object may be delivered. A *no-duplicates* service will detect and discard any duplicates.

Currently, two kinds of transport protocols are widely implemented:

- reliable, ordered, no-duplicates service. Example: TCP
- unreliable, unordered, duplicates-permitted service. Example: UDP

Current applications that need to communicate objects (i.e., images, video frames, files, sound bites) choose between classic transport services that provide either a reliable, ordered service or one that does not guarantee any ordering or reliability. However, many applications require services in between these two extremes—i.e., *partial-order service* and/or *partial-reliability* service.

Neither of the two transport services described above is, by itself, a good fit with the requirements of multimedia. Some objects in a multimedia presentation (e.g., control objects, text titles) may require reliable service (provided, e.g., through retransmission), while a best-effort service may be more appropriate for audio samples or video frames. Objects destined for the same display window may have an ordering requirement, while objects destined for different windows may arrive in many different orders.

Clearly, there exist many other possibilities for order and reliability besides these "all or nothing" services. For example, suppose we want to communicate n objects. We can represent the various possibilities for order using a partial order P over the set $[n] = \{1, 2, \ldots, n\}$, where $x \prec y$ in P signifies that object x must be delivered to the receiving application prior to object y. We can represent the various requirements for reliability by using a reliability vector $R = \langle r_1, r_2, \ldots, r_n \rangle$ where each r_i indicates the level of reliability required for object i. Each value of r_i gives some indication of how hard the transport protocol should work to retransmit object i if it is lost. For example, in the simplest case, we may define that $r_i = 1$ signifies that object i may be lost, and $r_i = 0$ signifies that object i may not be lost. In a more elaborate protocol, we could define the reliability vector over a set of *reliability classes*. For example, a class $\mathit{Retransmit\_until}(x)$ might indicate that an object should be retransmitted until time x, but that after that time it should be declared lost. Other reliability classes might specify that forward error correction is to be applied.

A *partial-order*, *partial-reliability*, *no-duplicates* (PO/PR) *transport protocol* is a transport protocol that allows the user to provide a partial order P and reliability vector R to specify the precise level of service required. *Partial Order Connection* (*POC*) is an example of such a protocol [1]. Another example of a PO/PR transport protocol is presented in RFC1693 [2], which describes how partial order and partial reliability could be added in a backwards-compatible fashion to a stream protocol such as TCP. PO/PR protocols provide a *PO/PR service* to the application layer.

Previous research using analytic and simulation models for POC compares the throughput and average packet delay of POC to that of an ordered, reliable service [3]. This analysis has shown that there is the potential for performance gains in terms of reduced delay, increased throughput, and reduced buffer requirements.

However, like most modeling studies, our analytic and simulation models of POC rely on several simplifying assumptions—assumptions that may limit the usefulness of our results in practice. In addition, the degree and significance of the performance gains is dependent on the partial order used. To put our results in a realistic perspective, it is necessary to (1) choose a specific application that can motivate the choice of partial order and partial reliability characteristics that are modeled, and (2) validate our analytical and simulation results with empirical performance measurements from a protocol implementation.

We believe that multimedia document retrieval with graceful degradation may be a suitable application for POC; this is motivated in the next section. We are currently in the early stages of designing and constructing a prototype multimedia document retrieval system based on a PO/PR transport service. Our success or failure in this effort will be crucial to the acceptance of partial order as a useful quality-of-service parameter at the transport layer.

3 Graceful Degradation: Motivation

Many multimedia documents have a temporal dimension; once they are called up for display, they are played out according to some schedule. Presentation according to this schedule progresses until some event occurs which stops or resets it (for example, a user presses a "pause" button, or reaches an interaction point.)

Typically, a multimedia workstation with sufficient processing capacity (i.e., CPU and memory) can present the document in compliance with this schedule, provided that the channel that delivers the information from its storage location is ordered and error-free. However, suppose the document is stored on a remote file server, and the channel delivering this information is a network connection. In this case, network errors may wreak havoc with attempts to present the document correctly.

We propose that in such situations, it may be appropriate to provide for "graceful degradation" of the presentation of the multimedia document. This is based on a recognition that in many multimedia documents, not all objects have the same importance, nor do they have the same quality-of-service requirements. Some objects are essential to document content, while others are "nice to have," but completely optional.

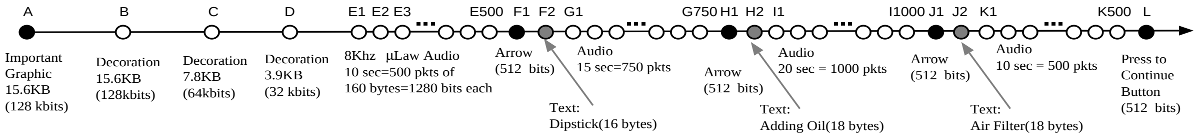
A
B
C
D
E1 E2 E3
E500
F1
F2
G1
G750
H1
H2
I1
I1000
J1
J2
K1
K500
L
Important Graphic 15.6KB (128 kbits)
Decoration 15.6KB (128kbits)
Decoration 7.8KB (64kbits)
Decoration 3.9KB (32 kbits)
8Khz µLaw Audio 10 sec=500 pkts of 160 bytes=1280 bits each
Arrow (512 bits)
Text: Dipstick(16 bytes)
Audio 15 sec=750 pkts
Arrow (512 bits)
Text: Adding Oil(18 bytes)
Audio 20 sec = 1000 pkts
Arrow (512 bits)
Text: Air Filter(18 bytes)
Audio 10 sec = 500 pkts
Press to Continue Button (512 bits)

Figure 1: Example document as a stream

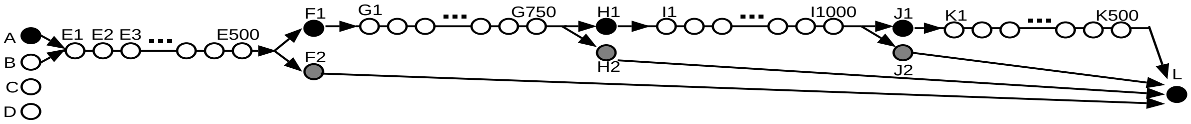

Figure 2: Example document as a partial order

4 Graceful Degradation: Example

Consider a system where multimedia documents are being retrieved from a remote server and displayed in real-time as the contents arrive at a user's workstation. Suppose that we desire to retrieve and display a document which describes three routine maintenance procedures that should be performed on a car: checking the oil, adding oil, and changing the air filter. Figure 1 illustrates the objects that might be present in such a document, as well as the sequence in which they might be transmitted over the network for display.

First, four graphic objects are placed on the screen: a picture of the engine (object A, which is critical to the presentation content), along with three objects (B,C,D) that are merely decorative. Next, an audio sequence plays out (E,G,I,K) that describes the three maintenance procedures.

Synchronized with this audio, arrows pointing to the parts of the engine affected (F1,H1,J1) appear at the appropriate times, along with text labels that describe the engine parts (F2,H2,J2). Finally, a "Press to Continue" appears (L).

A plan for graceful degradation of this document is represented by Figure 2. This diagram illustrates a partial order (represented as a precedence graph) and a partial reliability vector (represented by the various colors of the circles representing the objects.) The **black** circles represent objects which may not be lost; these objects will be retransmitted until they arrive. For each of these objects, if necessary, the presentation will be stopped until the critical object arrives and all predecessors have been delivered or declared lost. In the given example, the "critical" diagram of the engine is such an object, as are the arrows pointing to the various engine parts. It is assumed that the points in the audio track that are separated by these arrow objects represent places where a brief pause for retransmission of an object would be acceptable (e.g., a pause between complete sentences.) The **white** circles represent objects for which loss may be tolerated. These objects will be transmitted on a best effort basis, and the presentation will not stop if they do not arrive. If they have not arrived by the time they are scheduled for presentation, they are simply declared lost. This class includes the individual 1/50 second audio objects and the decorative graphics. The **gray** circles are objects which are valuable for some period of time, but subsequently are of no value; furthermore, they are not important enough to hold up the presentation if they do not arrive on time. The text labels at the ends of the arrows fall into this class, presumably because they are redundant—the audio track will also identify the engine parts as they are indicated by the synchronized arrows. We would like to retransmit them any time up until the user presses the button (L); after that, they are useless and should be dropped.

Consider what may happen to this presentation using three alternatives at the transport layer: UDP, TCP, and a PO/PR protocol such as POC. We will assume that a throughput of 128kbps is available, so there is clearly sufficient bandwidth to send the document and display it in real time. Suppose that the underlying network has a non-negligible loss rate such that it is expected that at least one network packet will be lost during the presentation.

Under UDP, we run the risk that certain critical objects (e.g., the picture of the engine, or the button that allows the user to continue) may be lost. In this case, the user of the program is likely to experience severe frustration. Unless the application provides partial reliability itself, graceful degradation is not possible: UDP has no features to support it.

Under TCP, we will retransmit any missing objects, even if they are objects for which loss may be tolerated. Furthermore, because the service is an ordered stream, any objects following the lost object will be delayed until the retransmission is successful. Suppose that one of the objects in the middle of one of the audio sequences (e.g., E250) is lost. In this case, there will be a gap in the playback of this audio—in the middle or a sentence, or possibly even in the middle of a word—while packet E250 is retransmitted. Our measurements of TCP retransmissions on wide-area Internet routes indicate that this gap could be as long as half a second.

Under POC, however, we have the flexibility to meet our goals. If any of the black objects are lost, the presentation schedule will stop until the objects can be retransmitted. Since this occurs only at logical stopping points in the audio track, and only for essential objects, the effect of this on the user is minimal. If audio objects are lost, each one is sufficiently small that the user is unlikely to object to the loss in audio quality; furthermore, since they are loseable, the presentation will not grind to a halt while a retransmission takes place. Loss of a "loseable" object will not cause delay in the delivery of other objects. Finally, the loss of a gray object will not delay subsequent audio objects; in fact, assuming there is adequate idle bandwidth, a lost gray object can likely be retransmitted in time to make an appearance while it is still useful.[1]

[1] For reasons of space, we omit certain details regarding the scheduling of retransmissions vs. original transmissions. It would not be good, for example, for a retransmission of the gray object to occur in the middle of a stream of audio packets if the effect was to cause a discontinuity in the audio playback.

5 Synchronization Issues

In designing a prototype document retrieval system based on POC, we are faced with a two important and related design decisions:

1. selecting an existing (or designing a new) formal description technique for specifying loss, order, and synchronization requirements for audio, video and still image objects (e.g., text, geometric, and bit-mapped images) within a multimedia document.
2. designing the protocol used by the client and server at the application layer that allows the client to provide the specified synchronization, order, and loss, given that the underlying transport provides PO/PR service.

It is our impression from the volume of research in this area that various techniques for synchronization specification and implementation have been proposed and successfully implemented; in fact, many such techniques will likely be reviewed in the course of this workshop. However, it is likely that current techniques are designed on the assumption of current transport protocols in which PO/PR service is not an option. We propose, therefore, to consider the body of work on synchronization from a new perspective. There are several key questions we hope to address through the construction of a prototype system:

1. What currently proven or proposed synchronization specification and implementation techniques might integrate well with a PO/PR service?
2. What new approaches to synchronization might be possible if they are implemented on top of PO/PR service?
3. Would a synchronization technique layered on top of a PO/PR service provide advantages in terms that are meaningful for the human viewer of the document (for example, better graceful degradation of document quality in the presence of network errors?)
4. Would a synchronization technique layered on top of a PO/PR service provide advantages in terms of ease of programming? That is, suppose the programmer of a multimedia document retrieval system can assume (because POC is used) that objects will arrive from the network according to a certain partial order, and will be dropped and/or retransmitted according to certain reliability classes (possibly related to timeliness). Does this help to simplify the task of coding the synchronization routines in this system?
5. What additional reliability classes or other features might need to be added to POC to make it more useful for synchronization?

References

[1] P.D. Amer, C. Chassot, T. Connolly, M. Diaz, and P. Conrad. Partial order transport service for multimedia and other applications. *IEEE/ACM Trans on Networking*, 2(5):440–456, Oct 1994.

[2] T. Connolly, P.D. Amer, and P. Conrad. Internet RFC1693: An Extension to TCP: Partial Order Service. Internet Request for Comments RFC1693, Nov 1994.

[3] R. Marasli, P. D. Amer, P. T. Conrad, and G. Burch. Partial order transport service: An analytic model. In *Ninth Annual IEEE Workshop on Computer Communications*, Marathon, Florida, October 1994.