Chapter 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

We conclude the dissertation with a summary of our key results (Section 7.1), an assessment of their applicability to other problems (Section 7.2), and a list of suggestions for future study (Section 7.3).

7.1 Summary of main results

Our summary of results in divided into three sections. Section 7.1.1 provides a general discussion of what we claim as our most significant result: the empirical demonstration that PO/R service can provide performance benefits *in practice*. Section 7.1.2 summarizes specific findings from our performance study. Section 7.1.3 summarizes the other important results in this dissertation.

7.1.1 PO/R service can provide benefits in practice

We begin with the most significant result of the dissertation:

We present empirical evidence that partially-ordered /reliable (PO/R) transport service can provide performance benefits over ordered/reliable (O/R) transport service in practice, for a concrete application.

The presentation of this evidence represents a major step forward in the study of partially-ordered transport service because these results remove two limitations of previous work. Prior to this dissertation, performance benefits from partially-ordered/reliable (PO/R) service had been shown:

- only through simulation and analytic modeling
- only in terms of improvements in throughput, delay and buffer utilization for a normalized time-scale, with no *specific* application or end-user context in which to judge the significance of these improvements (Marasli, 1999b).

Our data, which shows

- a specific application (our ReMDoR multimedia document retrieval system) and
- a set of network conditions where PO/R service is useful,

provides a concrete starting point from which to pursue further investigation. Further hypotheses about the performance of PO/R service for other applications and other network conditions can be developed from the analytic models, or extrapolated from our empirical data, and then tested through experimentation.

Specifically, our data provides:

> *a set of network conditions under which unordered/reliable and partially ordered/reliable service can provide perceptible performance improvements over ordered/reliable service in terms of progressive display.*

Under these conditions, unordered and partially ordered service provides access to the same information earlier than ordered service, with improvements ranging from a few seconds, to tens of seconds. Experiments **N1**, **N2**, and **R1**–**R4** illustrate such benefits for the following conditions:

- sliding window flow control, with no congestion avoidance features.

- Bernoulli packet losses
- constant bitrate and propagation delay, yielding a total roundtrip delay varying only due to processing delays and queuing delays at a single intermediate queue
- a range of bitrates from 2.4kbps to 128kbps, loss rates from 5% to 20%, and propagation delays ranging from zero to 500ms.
- documents that include images in parallel with one another, and images in parallel with audio streams.

Experiment **R5** shows these benefits under a different set of conditions:

- sliding window flow control, with TCP-like congestion avoidance features (i.e., the KX3 protocol using Jacobson's slow start and congestion avoidance mechanisms.)[76]
- Bernoulli packet losses
- constant bitrate and propagation delay, yielding a total roundtrip delay varying only due to processing delays, and queuing delays at a single intermediate queue
- a bitrate of 512kbps, loss rates of 5% and 10%, and a one-way propagation delay of 250ms.
- a document including 10s of images in parallel and sequence, a continuous audio stream, and complex synchronization relationships among the multimedia objects.

In addition, our data also show that PO/R service performs as well as—and in some cases better than—O/R service with respect to audio performance, reducing the number and length of interruptions in audio playback under moderate packet losses (around 10% one-way packet loss.) While it remains to be shown in future work whether the improvements in these audio metrics translate into perceptible

---

[76] Formal verification that implementation meets the criteria of *TCP-friendliness* is ongoing work being undertaken by the author.

performance improvements in terms of audio, we *have* shown that some degree of measurable improvement in objective audio metrics is obtainable.

7.1.2 Other results: transport service

In the area of transport service, we have made several contributions:

- A proposed data-preview (buffer access) feature, providing the application layer access to data that is buffered in the transport layer due to delivery order requirements (Section 2.7).
- A new proposed semantics for the PR reliability class, designed to use the partial order and synchronization information to detect and take advantage of opportunities for extra retransmissions that would not ordinarily be feasible. (Section 2.8)
- A proposed ADN-cancel feature, allowing the application layer to cancel delivery of ADUs in the transport layer pipe that have become useless to the receiver (Section 2.9)
- UTL, a library for experimentation with innovative transport services, providing a framework for rapid prototyping of transport layer implementations, and experimenting with application performance over a wide range of transport protocols (Chapter 3).
- A novel solution to user-space transport protocol CPU scheduling involving cooperative multitasking (Section 3.5.7)
- Efficient algorithms for PO/PR transport protocol processing that use the transitively reduced adjacency list representation of the partial order to reduce the computational complexity of protocol operations. (Chapter 6.)

7.1.3 Other results: multimedia systems

In the area of multimedia systems, we have the following results:

- An argument that graceful degradation should be incorporated in multimedia authoring systems, (Sections 1.5, 2.8) and performance results to support this argument, showing that partial order service can speed up progressive display of pixels, and reduce the number and duration of audio interruptions (Sections 5.2 through 5.7)

- ReMDoR: A prototype system for evaluating graceful degradation of temporal multimedia documents, including an authoring language, document compiler/scheduler, client and server. (Section 1.6.2, Chapter 4, Section 6.3).

- A method for unifying the notion of Object Composition Petri Net based synchronization with the notion of partial order delivery, thus providing a theoretical foundation for explicit release synchronization (Section 2.6).

- An argument for the benefits of integrating coarse-grained multimedia synchronization with transport protocol functions via explicit release, and the proposed semantics for the POCv2 PR reliability class, along with foundational work towards evaluating this argument with experimentation. (Sections 2.8, 6.8).

7.2 Applicability of our results to other applications

In this investigation, we examine only one application for PO/PR service, namely multimedia document retrieval. Using PO/PR service to support multimedia document retrieval was first proposed in (Amer et al., 1994), and developed further in (Conrad et al., 1996)). Until quite recently, the only other concrete application proposed for PO/PR transport service was the transmission of results of database queries (RFC1693). However, early in 2000, serious work began to be advanced on a new protocol, called the Stream Control Transmission Protocol (SCTP) that incorporates features for partial order (Stewart et al., 2000) In this section, we comment briefly on both of these potential applications.

7.2.1 Application of results in this dissertation to work on SCTP

SCTP has been developed within the Signaling Transport working group (sigtran) of the Internet Engineering Task Force (IETF). The primary motivation was to allow the upper layers of SS7, a protocol suite used by telephone companies for controlling network elements (e.g., switches) to operate over IP. SS7 is an example of what protocol designers sometimes call a “stovepipe”; that is, it is a protocol suite where the upper layers are tightly coupled with the transport, network, link and physical layers. Stovepipe designs are often efficient in terms of processing and bandwidth, but are difficult to migrate into new environments, or adapt to changes in technology. Because of the growing interest in Voice-over-IP networks, it has become desirable to consider breaking the stovepipe of SS7, and allowing the upper SS7 layers to run over IP’s unreliable datagram service. This is the primary purpose envisioned for SCTP.

However, it turns out that one of the needs that the upper layers of SS7 present is to set up multiple ordered, parallel reliable data streams within a single connection (or, in SCTP terminology, a single association.) Thus, SCTP provides a capability similar to the stream capability proposed within POCv2. Thus SCTP provides a restricted form of partial order service. While SCTP does not allow for complex partial orders, as does POCv2, one can imagine implementing a thin layer on top of SCTP that would result in a protocol equivalent or nearly equivalent to POCv2s PO/R service.

Furthermore, while SCTP was motivated by the need to transmit SS7 control data, the results in this dissertation indicate that the partial order features of SCTP may be able to provide significant benefits over UDP and TCP as a transport service for multimedia data. One crucial aspect of the SCTP design is that it is incorporating TCP-friendliness right from the beginning, under the supervision of many of the strongest advocates for TCP-friendliness in the IETF research community. Therefore, we envision a tie-in with SCTP research as a strong possibility for future work with partial order transport, and its application to multimedia systems.

Specifically, we propose experiments that would compare the operation of ReMDoR (or a multimedia document retrieval system with equivalent capabilities) over SCTP vs TCP, along the same lines as the experiments in Chapter 5. Such experiments would be of value for two reasons. First, the exercise of designing a multimedia system over SCTP would highlight whether SCTP's capabilities are sufficient to provide the performance benefits for multimedia that this dissertation has demonstrated are obtainable from partial order service. Such an investigation would be particularly timely in the immediate future, since SCTP is still currently in the experimental phase, and changes to the protocol would be more likely to be accepted than after SCTP has been incorporated into many commercial products. Second, the benefits of partial order service are more likely to be accepted by the broader Internet community after they have been demonstrated over a standards-track protocol with broad IETF support from many academic and commercial labs, rather than an experimental protocol from a single academic research lab, i.e. POCv2.

7.2.2 Applying partial order to the results of SQL queries

As promised in Chapter 5, we briefly comment here on the applicability of partial order service to the retrieval of the results of SQL queries, as proposed in RFC1693. Our premise is that there are three cases for an SQL query result. In the first case, the query result is small enough that the entire result can be visually scanned by a human being with a few seconds or minutes, and the goal is to present this information to a human being for browsing. In this case, the problem is essentially a special case of multimedia document retrieval. The goal is to provide a visual representation of the data as quickly as possible—that is, the application is providing progressive display. The second case is that the data is to be written to a file. This case is essentially a file transfer problem, and unordered reliable service suffices (such as first proposed in the so-called *Reliable Datagram Protocol* of RFC908, RFC1151). The third case is that the data is to be fed to a consumer process as part of a distributed computation. This latter case is the most interesting, but requires a specific model of the computation if one is to evaluate whether partial order delivery would provide any speedup for the overall system.

7.3 Future Work

Each chapter contains numerous suggestions for future work. To conclude the dissertation, we summarize these and in some cases, expand on our suggestions.

7.3.1 Experimentation with PO/PR service

Given our results that PO/R service can provide perceptible benefits, it is now interesting to investigate whether PO/PR service can improve significantly upon the benefits provided by PO/R service. Here is an example of a hypothesis that could be tested via experimentation with PO/PR service:

- PO/PR service can provide a useful tradeoff between delay and image quality in temporal multimedia documents containing network-conscious images.

This hypothesis could be tested by placing a series of partially reliable network-conscious images in a chain along with narration and pauses. After a fixed amount of time, the remainder of any incomplete image could be discarded. Part of the difficulty in testing such a hypothesis is that it is vague: it is difficult to state precisely what constitutes a “useful tradeoff.” It may be difficult to investigate this hypothesis without resorting to subjective opinion scores, since different viewers may have different preferences regarding the tradeoff between loss and delay. However, certain quantities can be objectively measured, including progressive display of both pixels and objects (which captures delay), and the rate at which pixels, and objects are declared lost. By varying both the loss rate, and the length of the pause during which the receiver waits for pixels for each image, we can objectively investigate the degree to which reliability and delay can be traded-off. Other possible hypotheses regarding PO/PR service include those related to audio performance at moderate loss rates, where the number of dropped cells can be traded-off with the duration of interruptions in the stream

7.3.2 Further experimentation with PO/R service

There are also additional directions to be taken with the PO/R experiments presented in this dissertation. The first is to begin conducting experiments over a real Internet connection, rather than a simulated one—specifically, a connection between the University of Delaware and Temple University. Such experiments will provide data that will be useful in determining the extent to which PO/PR service provides benefits under real network conditions. Undertaking this work will first involve completing the testing of Linux ports of both UTL and ReMDoR. This work is already underway at Temple University.

Second, it would be interesting to repeat some experiments using the Gilbert loss model, rather than the Bernoulli loss model. The Gilbert loss model may more accurately model burst packet losses resulting from drop-tail queuing policies in internet routers. (Section 5.1.4)

Third, it would be valuable to repeat Experiments **N1** and **N2**, and **R1–R4** using KX3. Of particular interest is the effect of receiver window size on throughput, and the interaction between the TCP-friendly features of KX3 with PO/R service. (Section 5.3.7) As suggested in Section 5.4.4, since TCP's congestion avoidance mechanisms dynamically change the window size as a response to congestion, an interesting topic for future work would be to investigate the effect of these window size changes on the performance difference between PO/R and O/R service. Such an investigation might plot measured RTT, effective window size, and the performance differential as a function of time.

Fourth, we suggest pursuing human factors studies focused on the experiments that seemed to show the most promising results based on the objective metrics, to test our hypothesis that these results are perceptible and significant in terms of end-user perceived performance (Section 5.3.4).

Finally, we note that the experiments reported in this dissertation cover a small range of multimedia documents. To investigate further whether the extent to which PO/PR is useful for multimedia documents in practice, it would be interesting to work with documents created by graphic designers rather than by computer scientists. It would first be necessary to provide some more intuitive way to create multimedia documents that can be retrieved via PO/R and/or PO/PR protocols. A more intuitive interface could be provided either by enhancing an existing authoring system with capabilities to create such documents, or by providing a more intuitive graphical user interface for the creation of ReMDoR documents. Then, such a tool could be provided to a group of graphic designers. The resulting documents could be used as the basis for performance experiments. Perhaps also, experiments could be done on hand-tuned versions of documents similar to the ones created by the artists, to investigate ways the tools could help to encourage authoring for graceful degradation. Experiments with a larger corpus of documents would also facilitate investigation of the correlation between performance gain and partial order density (Section 5.3.7).

In addition to these broad directions for future experiments, Section 5.6 contains several suggestions for particular parameter values that may be of interest for a future round of experiments with audio performance.

7.3.2 Other future work

Future Directions in Multimedia Systems

- Investigation of the Two-color Petri-net delivery semantics proposed in Section 2.6.4 to provide a better integration of Object Composition Petri Net synchronization and explicit-release synchronization with partially reliable transport service.
- Human factors studies to assign Mean Opinion Scores to reliable playback of audio with interruptions, and correlate these scores with our three proposed audio metrics (Section 5.6.1)

Future Work on ReMDoR

(See also: Section 4.9)

- To enable wider experimentation, porting the X-Windows version of the ReMDoR client and server to the Linux operating system, and/or implementing a portable Java version of the client
- Adding capabilities for multiple audio streams, hyperlinks, and video

- Improvements to the user interface, including bookmarks, forward, backward, history, and view source options.

- Adding the ADN-cancel feature to ReMDoR (section 2.9)

Future Directions in Transport Protocol Design

(See also: Section 2.10)

- Implementation of the *data preview* feature proposed in Section 2.7, along with performance experiments to investigate whether this feature can reduce delay for objects requiring computationally intensive presentation layer processing.

- Investigation of multiple reliability classes for POCv2 stream objects (Sections 2.8.4, 6.8.1).

- Investigation of the ADN-cancel feature. For example, this feature could be incorporated into the ReMDoR browser along with hyperlinks. Response time for hyperlinks could then be compared with the feature enabled, and disabled. Also, an investigation of the tradeoffs of search-trees vs hashing for the maintenance of the ADN dictionary is proposed. (Section 2.9)

- Experiments to compare the penalty of message-service vs. byte stream service. (Section 3.5.4)

- Research into what kind of congestion avoidance really is suitable when some of the loss is due to noise, and other loss is due to congestion, and simulation of PO/R and PO/PR service in that context.

- Further analytic studies along the lines of (Marasli et al., 1996b, 1997a, 1997b, 1998), but focused particularly on the case of parallel ordered streams, as in SCTP. In particular, it was mentioned in Section 5.3.7 that increasing the number of parallel streams reduces the probability that all streams become blocked waiting for retransmission of an out-of-sequence packet. It would be useful to model this analytically, and determine the relationship between number of streams and potential performance gain under various conditions.

Future work related to UTL implementation

(See also: Sections 2.10, 3.9)

- Full formal evaluation of the TCP-friendliness of KX3. This may involve, for example, simulation of a large number of TCP flows and one or more KX3 flows, to investigate whether these flows share the bandwidth fairly.

- Making UTL thread-safe, and taking advantage of POSIX threads for background transport protocol processing. (Section 3.5.7)

- Investigating whether it is feasible or desirable to re-implement UTL within the X-Kernel framework.

- Addition of features for QoS, window size, and mechanism negotiation to UTL (Sections 3.5.6, 3.9)

- Addition of features for negotiation of multiple epochs for POCv2, as suggested in Sections 2.2.8, and 2.4.5

- Implementation of the SRL layer for UTL as described in Section 3.3.8.

- Investigating which UTL features might benefit from migration to the kernel. (Section 3.5.1)