An Analytic Study of Partially Reliable Transport Services (Extended Abstract) *

Rahmi Marasli Paul D. Amer Phillip T. Conrad

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

1 Introduction

Many applications such as video and audio can tolerate loss. When the reliability (i.e., loss rate) of the underlying network is worse than the application's tolerance for loss, one strategy that can be employed to improve reliability is *retransmission* of lost packets in the transport layer. Although widely rejected, the retransmission of continuous media (e.g., audio) is shown to be feasible by Dempsey [3]. Retransmission-based transport protocols can offer greater reliability than the underlying network, in exchange for less desirable values for other QoS parameters (e.g., delay). However, if the level of reliability provided by the transport protocol is more than the application really needs, other QoS parameters may suffer unnecessarily. For example, if a reliable service (i.e., zero loss) is requested when a service with 2% loss would suffice, the extra delay incurred to achieve zero loss may be unacceptable.

Therefore, to achieve the best tradeoff between reliability and other QoS parameters, *Partially Reliable* services have been proposed [1, 3, 4]. Partially reliable service fills the gap between reliable and unreliable service by allowing applications to specify controlled levels of loss. Since partially reliable service does not insist on delivering all the data, it provides higher throughput and lower delay than reliable service, and at the same time, it respects the loss tolerance of the application.

This extended abstract describes an analytic model of retransmission-based partially reliable transport service. This model is used to derive several results that may be helpful to designers of multimedia applications in determining how and whether to incorporate partially reliable transport protocols into their systems:

1. Given a set of network conditions, we determine the penalty in terms of throughput and delay that is incurred to achieve a given level of partial reliability. This helps determine whether or not it is appropriate to migrate some applications with known loss, delay and throughput requirements from using an unreliable transport service (e.g., UDP) to a partially reliable one.
2. We determine the network conditions under which partially reliable transport service provides performance improvements over reliable service, and obtain quantitative measures on the performance gains. This helps determine what is the penalty today's applications pay for using a fully reliable transport service (e.g., TCP) when a partially reliable service would suffice.
3. We compare the performance of two different transport protocol approaches to detecting and controlling losses (sender-based and receiver-based), and show conditions under which the sender-based method provides better performance.

Our model of partial reliability differs from previous models in that:

- Our model studies the effect of ack losses, an issue that is ignored in [3, 4]. We show that this may be a significant issue.
- Our model compares two different types of error detection and recovery: sender-based methods and receiver-based methods. The difference between these methods lies in whether Transport Sender or

*This work supported, in part, by the National Science Foundation (NCR-9314056), the US Army Communication Electronics Command (CECOM), Ft. Monmouth, and the US Army Research Office (DAAH04-94-G-0093, DAAL03-92-G-0070).

Transport Receiver is responsible for detecting and recovering from packet losses. References [3, 4] consider only receiver-based methods. We show that in some cases, sender-based methods offer better performance.

The rest of this abstract provides a summary of the basic model (Section 2), a summary of the main results (Section 3), and some suggestions concerning the interpretation of our results (Section 4).

2 Analytic Model

In this section, we present an overview of the analytic model for providing partially reliable service. A more rigorous definition will appear in the full paper. This analytic model is similar to the one in [5, 6] that models the Partial Order Connection (POC) protocol.[1] We use a three layer architecture which includes only the network layer, the transport layer, and the user application layer (see Figure 1.A).

The network is assumed to provide an unreliable service (i.e., loss is possible). A partially reliable transport protocol enhances this service by making just enough retransmissions to satisfy the loss tolerance of the application. In providing partially reliable service, the transport layer must first *detect* the lost packet, and then *decide* whether or not to *recover* it. The transport layer provides partially reliable service either by a sender-based or receiver-based method.

2.1 Sender-Based Model

In a sender-based method, Transport Sender takes a packet from User Sender, transmits the packet over the network, then sets a timer and buffers it. If the corresponding ack does not arrive within its timeout period, Transport Sender assumes that the packet is lost, and chooses whether or not to retransmit the packet. As packets arrive at Transport Receiver, they are immediately delivered to User Receiver and acknowledged (via selective acks.) In the model, it is assumed that User Receiver can consume packets as fast as they become deliverable, and there is no problem with running out of buffer space at Transport Receiver.

In a real partially reliable protocol, Transport Sender may decide whether to retransmit the packet based on some application specific knowledge; for example, a delay bound specified by the application. We model this through via α_s, the Transport Sender recovery probability. After a loss occurs (either a packet with probability p or an ack with probability q) and is detected, Transport Sender decides to retransmit the packet with probability α_s (See Figure 1.B). We choose this way of modeling retransmission because it facilitates comparison between the sender-based and receiver-based models.

2.2 Receiver-Based Model

The transport layer provides partially reliable service by a receiver-based approach as follows: Transport Sender takes a packet from User Sender, transmits the packet over the network, buffers it, and waits for a response from Transport Receiver. For each successfully received packet, Transport Receiver sends a selective positive ack (PACK) to Transport Sender. When Transport Receiver detects a lost packet (through either gap-detection or loss-timers [4]), it decides to recover the lost packet with probability α_r and requests the retransmission of the lost packet by sending a selective negative ack (NACK) to Transport Sender (α_r is called Transport Receiver recovery probability; see Figure 1.C). With probability $1-\alpha_r$, Transport Receiver decides not to recover the lost packet and sends a PACK in order for Transport Sender to release the corresponding packet from its buffers. Thus, when Transport Sender receives a PACK, it releases the packet from its buffers and takes a new packet from User Sender for transmission. In the case of a NACK, the corresponding packet is retransmitted by Transport Sender.

By assumption, if t_{pack} represents the packet transmission time, Transport Receiver can detect a lost packet immediately at time "t_{pack} + one way propagation delay" after the corresponding packet's transmission has started. This is an optimistic assumption since in practice, Transport Receiver must wait until the arrival of packet $i+m$ (where m is some number ≥ 1) before deciding that packet i is lost. By this assumption a PACK or a NACK is expected to arrive at Transport Sender at time t_{out} (where t_{out} represents the timeout period, and is equal to the round-trip delay, which for modeling purposes is assumed to be constant—see Section 2.3) after each packet transmission (unless the PACK or NACK is lost). This optimistic assumption biases our

[1] POC is a proposed transport-layer protocol that provides *partially ordered* and *partially reliable* service to its users. POC fills the gap between *reliable and ordered* (e.g., TCP) and *unreliable and unordered* (e.g., UDP) services [1, 2].

results to some extent in favor of the receiver-based method; yet, as we shall see, the sender-based method still provides better performance for all the performance metrics computed.

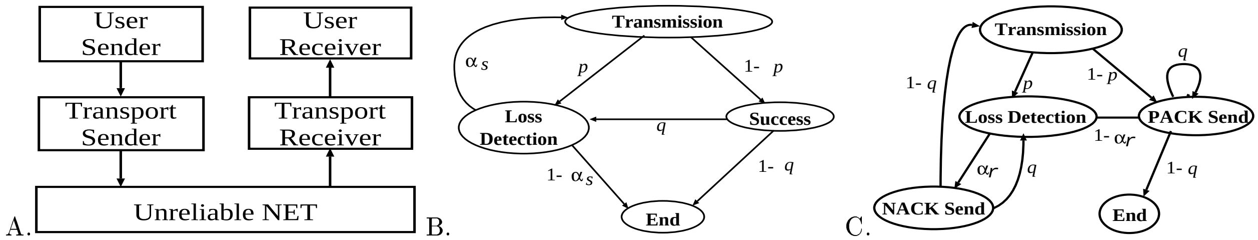

Figure 1: (A) Architecture and Diagram for (B) Sender-Based and (C) Receiver-Based Loss Detection and Recovery

2.3 Simplifying Assumptions

For purposes of modeling, some simplifying assumptions are made. It is assumed that User Sender submits constant size packets to Transport Receiver. It is also assumed that there are infinitely many packets waiting to be communicated at User Sender. User Receiver can always accept packets from Transport Receiver. In the network layer (called Unreliable NET), the loss of a packet or an ack is characterized by a Bernoulli process and a constant end-to-end network delay is assumed. The simplifying assumptions may affect the predictive power of the analytic model in terms of computing precise predictions for the target variables. However, we argue that the results obtained are still useful in *comparing* various types of service, and analyzing the *trends*, since we expect the effect of these assumptions to be similar across various levels of reliability (i.e., reliable, partially reliable, unreliable), and for both sender and receiver based methods. One exception to this is discussed in Section 4.

3 Main Results

In the full paper, we proceed from the models of the sender-based and receiver-based methods to derive formulae for various performance metrics. These formulae illustrate how these metrics are functions of the Transport Sender (or Receiver) Recovery Probability α_s (or α_r), and the packet and ack error rates (p and q, respectively). They also allow a comparison between the expected performance of the two methods.

The main performance metrics derived are:

- PLD, the packet delivery probability. This is the probability that a packet is delivered to its destination by the transport layer. PLD can also be seen as the probabilistic delivery guarantee provided by partially reliable service.
- λ_{US}, the transport layer admission rate. This is the rate at which Transport Sender accepts packets from User Sender.
- λ_{UR}, the throughput achieved. This is the rate at which the receiving application (i.e., User Receiver) gets data packets.
- *Delay*, the transport layer delay. This is the expected time for a packet to arrive to Transport Receiver, once it is given to Transport Sender by User Sender. It does not include any delay caused by buffering at Transport Receiver (e.g., for purposes of reordering data.)

For each of these metrics, the full paper includes:

- a discussion of the derivation of the formula,
- graphs plotted against α_s (or α_r, as appropriate), p and q, and
- a discussion of the significance of these results.

For purposes of this abstract, we merely state the main results for both sender and receiver, and highlight a few key issues. In addition to the notation defined above (p, q, α_s, α_r), the following notation is used in the remainder of this section: t_{out} represents the timeout period and is equal to round-trip delay. t_{delay} represents the one-way propagation delay. p_{succ} represents the probability of a successful packet transmission and ack transmission, i.e., $p_{succ} = (1-p)*(1-q)$.

3.1 Results for Sender-Based Method

(1) $$PLD = \frac{1-p}{1-p*\alpha_s}$$

(2) $$\lambda_{US} = \frac{1-(1-p_{succ})*\alpha_s}{t_{pack}}$$

(3) $$\lambda_{UR} = \frac{1-(1-p_{succ})*\alpha_s}{t_{pack}} * \frac{1-p}{1-p*\alpha_s}$$

(4) $$Delay = t_{delay} + \frac{p*\alpha_s}{1-p*\alpha_s} * t_{out}$$

- PLD does not depend on ack loss rate (i.e., q). Whether we lose *no* acks or *all* of the acks, delivery probability does not change. Thus, expression (1) shows that a sender-based approach can provide reliability guarantees regardless of ack loss level.
- PLD increases almost linearly as α_s increases. Having $\alpha_s = 0.8$ is enough to make delivery guarantees of higher than 97.8% at practical packet loss levels (i.e., $p \leq 0.1$).
- In expression (3), $\lambda_{UR} \geq \lambda_{Reliable}$ since $p \leq 1-p_{succ}$. This expression shows that partially reliable service provides throughput improvements over reliable service as long as there are ack losses in the network layer (i.e., $p \leq 1-p_{succ}$).
- For reliable service (i.e., $\alpha_s = 1$), $\lambda_{UR} = \frac{p_{succ}}{t_{pack}}$ and for unreliable service (i.e., $\alpha_s = 0$), $\lambda_{UR} = \frac{1-p}{t_{pack}}$. Therefore, the maximal throughput improvement by any partially reliable service is bounded by $\frac{(1-p)*q}{t_{pack}}$. For 10% loss level (i.e., $p = q = 0.1$) and 5.3% application loss tolerance (i.e., $\alpha_s = 0.5$ and $PLD = 0.947$), the throughput improvement by partially reliable service over reliable service is about 6%.
- In general, the throughput gain of partially reliable service over reliable service increases as an application's loss tolerance and ack loss rate increase.
- Expression (4) shows that *Delay* is independent of ack loss rate. Thus, a sender-based approach can provide *Delay* guarantees regardless of ack loss level. As expected, $Delay = t_{delay}$ for unreliable service and $Delay = t_{delay} + \frac{p}{1-p} * t_{out}$ for reliable service.
- Partially reliable service provides considerable delay improvement over reliable service even at the practical loss levels. For example, for $p = q = 0.1$, $t_{out} = 2 * t_{delay}$ and about 6% application loss tolerance (e.g., $\alpha_r = 0.5$), *Delay* can be as much as 9.6% lower by partially reliable service.
- The delay gain of partially reliable service over reliable service increases as an application's loss tolerance and packet loss rate increase.

3.2 Results for Receiver-Based Method

(5) $$PLD = \frac{(1-p)*(1-q*\alpha_r)}{1-(1-p_{succ})*\alpha_r}$$

(6) $$\lambda_{US} = \frac{(1-q)*(1-(1-p_{succ})*\alpha_r)}{1-q*\alpha_r} * \frac{1}{t_{pack}}$$

(7) $$\lambda_{UR} = \frac{p_{succ}}{t_{pack}}$$

(8) $$Delay = t_{delay} + \frac{p*\alpha_r}{1-(1-p_{succ})*\alpha_r} * t_{out}$$

- Unlike the delivery guarantee of the sender-based approach, PLD in receiver-based approach depends on not only packet but also ack loss rate.
- The sender-based approach provides slightly better reliability guarantees than the receiver-based one for all levels of the application's loss tolerance and network loss levels.
- The sender-based approach provides higher admission rate and throughput than the receiver-based one for all reliability levels (i.e., α_s and α_r values.) The advantage of the sender-based method over the receiver-based method increases with the ack loss rate.

- The sender-based approach provides lower delay than the receiver-based one at every application loss-tolerance level. But, the delay improvement of a sender-based approach over receiver-based one is not large at practical loss levels.

- In general, the performance improvements of a sender-based approach over receiver-based one increase with the ack loss rate.

4 Limitations of Model

The full paper contains a discussion of the various simplifying assumptions that are made, and the ways in which those assumptions may limit the usefulness of the results. For this abstract, we focus on only one issue: the effect of Sender Buffer Size on the comparison between sender-based and receiver-based methods.

Our results show that under the assumptions given, the sender-based approach provides improvements in delivery guarantee, delay, and throughput, particularly as the ack loss rate increases. If we look at the assumptions, except for setting the buffer size at the sender equal to the pipe size, ($Buf_S = \frac{t_{out}}{t_{pack}}$), they all define the ideal conditions to get maximal system performance for both sender-based and receiver-based approaches (e.g., no processing time at each side, infinite buffer space at the receiver, etc). Thus, except for Buf_S, the comparison seems to be fair.

However, the choice of Buf_S as the pipeline size (i.e., the bandwidth-delay product) biases the results in favor of the sender-based approach. This is because the receiver-based approach can increase its throughput and the admission rate by increasing the buffer size at the sender beyond the pipeline size. This is not possible with the sender-based approach (since for modeling purposes the timeout value t_{out} is taken to be the fixed round-trip delay.) Thus, the receiver-based approach can potentially be better in terms of admission rate and throughput if an arbitrarily large buffer size is used at the sender.

Nevertheless, the sender-based approach may have higher throughput in situations where there are ack losses, and for situations where it is not feasible to significantly increase the buffer size beyond the pipeline size (such as applications that serve a large number of clients). Furthermore, the sender-based approach provides better delivery guarantees and lower delay than the receiver-based approach regardless of buffer size.

5 Summary

This paper provides a model for a partially reliable transport protocol based on retransmissions. This model illustrates the tradeoffs that are possible between three QoS parameters (delay, throughput and delivery probability), and various levels of reliability. It also allows comparison of sender-based and receiver-based methods. For applications with known delay and loss tolerance (such as multimedia), such a model is helpful in determining whether, and to what extent to employ retransmission as a strategy for improving overall communication reliability.

References

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

[2] Thomas J. Connolly, Paul D. Amer, and Phillip T. Conrad. RFC-1693, An Extension to TCP: Partial Order Service.

[3] Bert J. Dempsey. *Retransmission-Based Error Control For Continuous Media Traffic In Packet-Switched Networks.* PhD Dissertation, University of Virginia, 1994.

[4] F. Gong and G. Parulkar. An Application-Oriented Error Control Scheme for High-Speed Networks. Tech Report WUCS-92-37, Department of Computer Science, Washington University in St. Louis, November 1992.

[5] Rahmi Marasli, Paul D. Amer, and Phillip T. Conrad. Partial Order Transport Service: An Analytic Model. (Submitted for publication).

[6] 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. IEEE.

[7] Jean Walrand. *Communication Networks: A First Course*. Aksen Associates, 1991.