

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 April 2026

S. Harper
Independent
5 October 2025

A Historical Analysis of Design Lessons from The Reliable Data Protocol
Version 1 and Version 2
draft-harper-rdp-analysis-00

Abstract

This document reviews and analyzes the history and experience of the Reliable Data Protocol. RDP was initially published as an experimental protocol in RFC 908 in 1984 and subsequently revised in RFC 1151 in 1991. RDP aimed to provide a message-oriented, reliable transport service to meet specific application requirements such as remote loading. This document summarizes the key technical experiences gained from RDP's experimental deployment, details the lessons learned regarding checksum performance and flow control mechanisms, and assesses the conceptual influence of its design principles on subsequent IETF transport protocols. The document is intended for historical archiving, providing a reference for early explorations of Internet transport protocols, and does not propose any new standards or deployment recommendations.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at <https://github.com/sphpr/rdp-info>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	2
1.1. Context and Objective	3
1.2. Document Scope and Methodology	3
2. RDP Design Philosophy and Early Innovation	3
2.1. Core Features: Message-Orientation and Optional Ordering	3
2.2. Pioneering Selective Acknowledgement	4
3. Technical Experience and Key Lessons	4
3.1. Checksum Performance and Engineering Lesson	4
3.2. Deficiencies in Flow Control Mechanism	4
3.3. Absence of Congestion Control	5
4. Conceptual Impact on Successor Protocols	5
4.1. Conceptual Precursor to SACK	5
4.2. Validation for Multi-Streaming and Unordered Delivery . .	5
5. Summary of Engineering Lessons Learned	6
6. Security Considerations	6
7. Conclusion	7
8. IANA Considerations	7
9. References	7
9.1. Normative References	7
9.2. Informative References	7
Acknowledgments	7
Author's Address	8

1. Introduction

1.1. Context and Objective

The Reliable Data Protocol (RDP) was released in 1984 via RFC 908, forming part of a series of early Internet experimental protocols. The protocol attempted to offer a transport layer service distinct from the existing Transmission Control Protocol (TCP), designed specifically for applications that require explicit message boundaries and are sensitive to latency, such as remote debugging and file transfer. Although RDP's initial version (RFC 908) was designated as experimental, and the protocol never progressed to an advanced stage of the IETF standardization process, the lessons learned from its experimental deployment hold lasting value for the Internet community. This document serves to formally record this history and to codify its technical lessons⁵.

1.2. Document Scope and Methodology

This document is intended to serve as an Informational memo to provide historical context for transport protocol designers. This analysis focuses on the original RDP specification (RFC 908) and its revision (RFC 1151). Key areas of experience examined are: first, the innovative design of message orientation and selective acknowledgment; second, the performance bottlenecks and solutions for the checksum algorithm; and third, the mechanism deficiencies found in the window-based flow control. This document does not advocate for the redeployment or standardization of RDP, but rather focuses on summarizing its historical contribution through the distillation of engineering lessons.

2. RDP Design Philosophy and Early Innovation

2.1. Core Features: Message-Orientation and Optional Ordering

The core of RDP's design was to provide a message-oriented reliable service, contrasting sharply with TCP's byte-stream model. This was achieved by explicitly defining message boundaries within the protocol segments, aiming to simplify application-layer parsing and processing logic. While the specific mechanism (such as a flag or length field) is detailed in the specifications, the intent was to offer a more natural fit for discrete data transfers. Furthermore, RDP allowed applications to choose whether ordered delivery was required. This early optimization philosophy permitted the protocol to avoid unnecessary buffering and latency in scenarios where strict sequencing was not mandated.

2.2. Pioneering Selective Acknowledgement

A key innovation of RDP was its adoption of an Explicit Acknowledgement (EACK) segment mechanism. This mechanism enabled the receiver to precisely inform the sender which specific out-of-order segments had been successfully received. This significantly reduced the unnecessary retransmission of all subsequent received segments caused by the loss of a single segment, which enhanced efficiency, particularly for bulk data transfer. The EACK mechanism represented an early, practical exploration of the concept of selective acknowledgment. However, the implementation of this mechanism also increased the complexity of state maintenance for out-of-order segments at the receiver.

3. Technical Experience and Key Lessons

3.1. Checksum Performance and Engineering Lesson

The first significant engineering challenge RDP encountered during experimental deployment was the performance of its original checksum algorithm. RFC 1151 notes that the checksum used in RDP V1 exhibited surprisingly variable computational costs across different host architectures and implementations¹⁰. In some cases, performance of optimized implementations differed by a factor of five on comparable hardware bases, severely impacting the protocol's throughput.

The variability in performance was deemed undesirable. Consequently, the revision in RDP V2 (1990) ultimately adopted the 16-bit one's complement sum algorithm used by TCP and UDP.

Lesson Learned: Performance measurement demonstrated that choosing an algorithm that is simple to implement and widely adopted can mitigate the unpredictable performance penalties associated with complex computation across varied CPU architectures, thus prioritizing engineering consistency over theoretical novelty.

3.2. Deficiencies in Flow Control Mechanism

RDP's window-based flow control mechanism revealed a serious design flaw. RFC 1151 explicitly states that RDP's flow control scheme does not allow the receiver to close the sender's window¹⁵. Specifically, if the receiver acknowledged segments immediately upon reception, even if its buffer was full, the sender would continue to transmit data based on the acknowledgment, potentially exceeding the receiver's processing capacity.

Technical Root Cause and Lesson: The practical consequence of this flaw was receiver buffer overflow and resource exhaustion. This RDP experience underscored that in a receiver-driven flow control mechanism, precise semantics for window withdrawal (reducing the window size) are fundamental requirements for protocol stability. The inability to communicate a zero or reduced window reliably allows the sender to overrun the receiver's resources, which can lead to protocol stack instability. The suggested solution in RFC 1151 (to only acknowledge segments after delivery to the application) was explicitly noted as untested at the time of publication.

3.3. Absence of Congestion Control

The limitations of RDP were compounded by its lack of a dedicated congestion control mechanism. While flow control is designed to protect the receiver's buffer resources, congestion control is necessary to protect the shared network resources. RDP's design primarily focused on receiver limitations, characteristic of early protocols developed before the critical need for global network congestion management was fully recognized. This systemic absence was a significant factor that precluded its large-scale adoption in the shared Internet environment.

4. Conceptual Impact on Successor Protocols

Although RDP did not achieve standardization, its core innovative concepts found clear conceptual resonance in subsequent transport protocol efforts.

4.1. Conceptual Precursor to SACK

RDP's EACK mechanism stands as one of the earliest implemented examples of selective acknowledgment technology within the IETF ecosystem. This practical experience provided a valuable reference for the subsequent evolution of the TCP protocol. The historical work of RDP demonstrated the efficiency gains offered by this mechanism in reliable transmission. This early experience conceptually foreshadowed the later SACK extension for TCP (e.g., RFC 2018) and the design of protocols like SCTP and QUIC.

4.2. Validation for Multi-Streaming and Unordered Delivery

RDP's design, which included support for explicit message boundaries and optional ordered delivery, conceptually aligns with the later multi-streaming and unordered delivery features found in protocols such as SCTP and QUIC. RDP's experience validated the application layer's practical need for these flexible transport semantics, guiding the design direction of subsequent transport protocols.

intended for complex application environments.

5. Summary of Engineering Lessons Learned

The design, revision, and experimental experience of RDP yield the following engineering principles for the IETF community:

Engineering Consistency Precedes Theoretical Innovation: The selection of a checksum algorithm should prioritize consistent performance and engineering simplicity across diverse hardware and implementations over theoretically marginal gains in error detection capability.

Flow Control Mechanisms Must Be Complete: It is essential to ensure that the receiver possesses a reliable and unambiguous mechanism to completely withdraw or limit the sender's transmission window, preventing receiver resource exhaustion. Flow control flaws can lead more directly to protocol stack failure than congestion control deficiencies.

Complex Features Require Cost Assessment: Although features like selective acknowledgment can enhance efficiency, the design must fully consider the implementation cost associated with receiver state maintenance and segment encoding complexity to ensure practical viability in deployment.

6. Security Considerations

RFC 1151 explicitly states that Security issues are not addressed in this memo¹⁸. RDP was designed in the mid-1980s, a period when Internet protocol design generally did not prioritize authentication, encryption, or message integrity. Consequently, RDP lacks the security mechanisms required by modern protocols.

As a historical record, this document makes no security proposals. Should RDP be considered for deployment in a modern network environment (which is not recommended), a comprehensive analysis of modern security requirements would be mandatory, including message authentication, integrity protection, resistance to replay attacks, and interoperability with established security frameworks such as IPsec or TLS/DTLS.

7. Conclusion

The Reliable Data Protocol (RDP) represents a valuable experimental attempt in the evolution of Internet transport protocols. It embodies early consideration of TCP's limitations, demonstrating foresight particularly in its message-oriented approach and use of selective acknowledgment. Concurrently, the flaws exposed in its checksum performance and flow control mechanisms provide an important historical case study for reinforcing rigor in subsequent IETF transport layer protocol design. This document completes the summary and archival of RDP's experience, intending to preserve this important historical reference for the Internet community.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

- [RFC1151] Partridge, C. and R. Hinden, "Version 2 of the Reliable Data Protocol (RDP)", RFC 1151, DOI 10.17487/RFC1151, April 1990, <<https://www.rfc-editor.org/rfc/rfc1151>>.
- [RFC908] Velten, D., Hinden, R., and J. Sax, "Reliable Data Protocol", RFC 908, DOI 10.17487/RFC908, July 1984, <<https://www.rfc-editor.org/rfc/rfc908>>.

9.2. Informative References

- [Partridge87] Partridge, C., "Private Communication", February 1987.
- [RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", RFC 2018, DOI 10.17487/RFC2018, October 1996, <<https://www.rfc-editor.org/rfc/rfc2018>>.

Acknowledgments

The authors wish to formally acknowledge the significant contributions of the original designers and implementers of the Reliable Data Protocol (RDP). Specifically, we recognize David Velten, Robert Hinden, and Jim Sax, who authored the initial experimental specification, RFC 9081. Their work represented a pioneering effort in transport protocol design, laying the groundwork for a message-oriented service in the nascent Internet.

We extend particular gratitude to Craig Partridge and Robert Hinden, who diligently revisited the protocol, summarized the critical findings from its experimental deployment, and authored RFC 1151 in 1990. This revision, which cataloged essential technical lessons concerning checksum performance and flow control deficiencies, provided invaluable, early engineering guidance to the wider Internet community.

Furthermore, acknowledgment is due to the members of the End-to-End Research Group who participated in RDP's testing and offered insights, including the proposed solution for the flow control problem.

Their collective efforts, even in the context of an experimental protocol that did not reach full standardization, continue to inform and influence the design of modern, more sophisticated transport layer technologies.

Author's Address

Sophie Harper
Independent
Email: sphpr@proton.me