Network Working Group
Request for Comments: 1151
Updates: RFC 908

C. Partridge
BBN Systems and Technologies
R. Hinden
BBN Communications Corp.

April 1990

Version 2 of the Reliable Data Protocol (RDP)

Status of this Memo

This RFC suggests several updates to the specification of the Reliable Data Protocol (RDP) in RFC-908 based on experience with the protocol. This revised version of the protocol is experimental.

Distribution of this memo is unlimited.

Introduction

Experiments in 1986 and 1987 turned up some ambiguities and problems with the RDP specification. At the time, it was hoped that the authors might find the time to revise the entire RDP specification to fix these problems, however given the limited demand for RDP implementations, the authors were never able to justify the time involved in revising the spec. This document lists the changes that we believe are appropriate to make to RDP version 1.

Readers are expected to be familiar with RFC-908.

Changes To The Protocol Header

There are three changes to the protocol header: the checksum algorithm has been changed, the port size increased, and the version number incremented. The new header format is shown in Figure 1.

The major discovery during the testing of the protocol is that cost of computing the the RDP checksum proved surprisingly variable; its performance was more heavily affected by the host's data representation than anticipated. Optimized checksum implementations on two comparable hardware bases gave performance that differed by a factor of five. Since the speed of the checksum is a key factor in the performance of the protocol itself, this variation caused a noticeable difference in throughput.

The wide variation in performance on comparable machines was felt to be undesirable, so the checksum has been changed. RDP now uses the 16-bit TCP checksum, which is specified on page 16 of RFC-793.

The 8-bit port size is probably too small to support a large range of applications. Accordingly, the port size is now 16-bits. Port numbers less than 1024 are reserved for well-defined applications. Allocable ports begin at port number 1024.

Finally, because the checksum and port size have been changed, the version number has been increased to 2.

```
       0             0 0   1         1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      +-+-+-+-+-+-+---+---------------+
      |S|A|E|R|N| |Ver|    Header     |
    0 |Y|C|A|S|U|0|No.|    Length     |
      |N|K|K|T|L| |   |               |
      +-+-+-+-+-+-+---+---------------+
    1 |         Source Port           |
      +---------------+---------------+
    2 |       Destination Port        |
      +---------------+---------------+
    3 |          Data  Length         |
      +---------------+---------------+
    4 |                               |
      +---    Sequence Number      ---+
    5 |                               |
      +---------------+---------------+
    6 |                               |
      +--- Acknowledgement Number  ---+
    7 |                               |
      +---------------+---------------+
    8 |           Checksum            |
      +---------------+---------------+
    9 |     Variable Header Area      |
      .                               .
      .                               .
      |                               |
      +---------------+---------------+
```

RDP Header Format
Figure 1

Minor Errors and Ambiguities

Some ambiguities and minor errors have been found in RFC-908. They are corrected in this section.

The value of the state variable, SND.UNA is treated inconsistently in the pseudo-code on pages 21-29. On page 12, SND.UNA is defined as "the sequence number of the oldest unacknowledged segment", and on page 21 it is appropriately set to the initial sequence number when the connection is opened. But on page 28, when an acknowledgement is received, SND.UNA is set to SEG.ACK, the sequence number being acknowledged, instead of SEG.ACK+1. A similar inconsistency occurs on page 26. The proper fix is to always set SND.UNA to SEG.ACK+1.

The pseudo-code on page 25 for the SYN-SENT state is incorrect. The first few lines cause all packets with the ACK bit set to be discarded, but later lines test the ACK bit. The test for the ACK bit should be placed after all the other tests. Also note that if the ACK bit is set, a RST segment is sent to reset the remote peer, but the local peer is left half-open. There is a similar problem in the SYN-RCVD state. The local peer should deallocate the connection record and close.

On page 24, the pseudo-code indicates that if non-data packets are received in the CLOSED state, a RST segment with SEG.ACK set to RCV.CUR should be sent. RCV.CUR is not defined in the CLOSED state. SEG.ACK should be set to SEG.SEG.

There is some inconsistency about how to handle a RST packet in the CLOSE-WAIT state. On page 24, the pseudo-code shows that a RST should cause the connection state to become CLOSED. Text on page 13 and the state diagram on page 10 suggest the connection state should stay in CLOSE-WAIT. The implementation should stay in CLOSE-WAIT.

On page 29, the pseudo-code for the OPEN state suggests that if a data packet is received in sequence, the acknowledgement packet should not contain EACKs. This is misleading. Implementations may include EACKs in the acknowledgement.

On page 18, it is possible to interpret the right edge as being either inside or outside the window. This results in a one segment difference in the window size. The proper interpretation is that the right edge is outside the window. In other words, the right edge is the first sequence number that cannot be sent or received and the total window size is 2*X, where X is the maximum number of outstanding segments.

One final problem is that RDP's flow control scheme does not allow the receiver to close the sender's window. As a result, if the receiver acknowledges segments when they are received the sender can easily send more data than the receiver is prepared to buffer. A solution to this problem (suggested by members of the End-2-End Research Group) is to only acknowledge a segment after it has been delivered to the application. This scheme, however, has not be tested.

Security Considerations

Security issues are not addressed in this memo.

Authors' Addresses

Craig Partridge
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, MA 02138

Phone: (617) 873-2459

EMail: craig@BBN.COM

Robert Hinden
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, MA 02238

Phone: (617) 873-3757

Email: HINDEN@BBN.COM