SANDIA REPORT
SAND2011-3500
Unlimited Release
Printed May 2011

Joint Architecture Standard (JAS) Reliable Data Delivery Protocol (RDDP) Specification

Mike Gardner, Richard Hunt, Justin Enderle, Daniel Gallegos, John Eldridge, and Jim Daniels

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-3500
Unlimited Release
Printed May 2011

Joint Architecture Standard (JAS) Reliable Data Delivery Protocol (RDDP) Specification

Michael Gardner
Embedded Radar Processing, 5348

Richard D. Hunt
Flight Embedded SW and Simulation, 5336

Justin W. Enderle
Wireless & Event Sensing Applications, 2664

Daniel E. Gallegos
Sensors and Embedded Systems, 2623

John M. Eldridge
Embedded Systems Engineering, 5632

James W. Daniels
Embedded Digital Subsystems, 5337

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0532

Abstract

The Joint Architecture Standard (JAS) program at Sandia National Laboratories requires the use of a reliable data delivery protocol over SpaceWire. The National Aeronautics and Space Administration at the Goddard Spaceflight Center in Greenbelt, Maryland, developed and specified a reliable protocol for its Geostationary Operational Environment Satellite known as GOES-R Reliable Data Delivery Protocol (GRDDP). The JAS program implemented and tested GRDDP and then suggested a number of modifications to the original specification to meet its program specific requirements. This document details the full RDDP specification as modified for JAS.

The JAS Reliable Data Delivery Protocol uses the lower-level SpaceWire data link layer to provide reliable packet delivery services to one or more higher-level host application processes. This document specifies the functional requirements for JRDDP but does not specify the interfaces to the lower- or higher-level processes, which may be implementation-dependent.

TABLE OF CONTENTS

1. INTRODUCTION 11
   1.1 Scope 11
   1.2 GRDDP Changes 11

2. DEFINITIONS 12

3. OVERALL FUNCTIONAL DESCRIPTION 13
   3.1 Multiplexed Logical Channels 13
      3.1.1 Channel Independence 13
      3.1.2 Transmit Priority 13
      3.1.3 Data Transmit Queue 13
      3.1.4 Urgent Messaging (Optional) 13
   3.2 Reliable Delivery 13
      3.2.1 Error Detection 14
      3.2.2 Packet Sequence Numbers 14
      3.2.3 Sequence Number Use 14
      3.2.4 Acknowledgment and Retransmit 14
      3.2.5 Retransmission 14
   3.3 Buffer Fragmentation and Reassembly 14

4. PACKET FORMAT 15
   4.1 Header 15
      4.1.1 Destination Address 15
      4.1.2 Protocol ID 15
      4.1.3 Packet Control 16
      4.1.4 Payload Length 17
      4.1.5 Channel Number 17
      4.1.6 Sequence Number 17
      4.1.7 Address Control 17
      4.1.8 Source Address 18
   4.2 Payload 18
      4.2.1 Secondary Header (Optional) 18
      4.2.2 DATA Packets and URGENTs 18
      4.2.3 ACK and CONTROL Packets 18
   4.3 Trailer 19
      4.3.1 CRC 19

5. CHANNEL OPERATIONS 20
   5.1 TEP Parameters 20
   5.2 TEP States 20
      5.2.1 Channel Closing Process 21
   5.3 Logical Connections 25
      5.3.1 OPEN/RESET Command 25
      5.3.2 CLOSE Command 25
      5.3.3 CONTROL Timer Cancellation 25
      5.3.4 CONTROL Timer Expiration 25
   5.4 Transport Channel Connection 25

5.5 Receive TEP Operations 26
5.5.1 Sliding Window 26
5.5.2 Sliding Window Size 26
5.5.3 Sliding Window Range 26
5.5.4 Window Advance 26
5.5.5 Packet Acknowledgment 26
5.5.6 Packets with Errors 26
5.5.7 Out of Window Sequence Number 26
5.5.8 Duplicate Sequence Number 26
5.5.9 URGENT Acknowledgment (If Implemented) 27
5.5.10 URGENT Delivery Order (If Implemented) 27
5.5.11 URGENT Delivery Priority (If Implemented) 27
5.5.12 CONTROL Packet Sequence Number 27
5.5.13 OPEN/RESET Command Processing 27
5.5.14 CLOSE Command Processing 27
5.5.15 Packets Pending Delivery 27
5.5.16 OPEN/RESET Command Report 27
5.5.17 CLOSE Command Report 27
5.6 Transmit TEP Operations 28
5.6.1 Transmit TEP ACKs 28
5.6.2 Transmit TEP Sequence Number Allocation 28
5.6.3 CONTROL Packet Sequence Number 28
5.6.4 Transmit Window 28
5.6.5 Unacknowledged Packets 28
5.6.6 Transmit Window Start 28
5.6.7 Transmit Window Advance 28
5.6.8 Packet Retransmit 28
5.6.9 Retry CONTROL 28
5.6.10 Timeout Start 29
5.6.11 URGENT Packet Transmission (If Implemented) 29

6. REFERENCES 30

APPENDIX A. Joint Architecture Standard (JAS) Reliable Data Delivery Protocol (RDDP) Specification Program Supplement 31

FIGURES

Figure 1. JRDDP packet format. 15
Figure 2. Nominal case. 21
Figure 3. Delayed case. 23
Figure 4. Worst case. 24

TABLES

Table 1. JRDDP Modifications. 11
Table 2. Buffer Sequence Flag Values. 16
Table 3. Packet-Type Values. 16
Table 4. TEP Parameters. 20
Table 5. TEP States. 20

ACRONYMS

ACK	Acknowledgment
CRC	Cyclic Redundancy Check
GRDDP	GOES-R Reliable Data Delivery Protocol
JAS	Joint Architecture Standard
JRDDP	JAS Reliable Data Delivery Protocol
NASA	National Aeronautics and Space Administration
MTU	Maximum Transmission Unit
SLA	SpaceWire Logical Address
TEP	Transport End Point
UML	Unified Modeling Language

1. INTRODUCTION

The Joint Architecture Standard (JAS) program at Sandia National Laboratories requires the use of a reliable data delivery protocol over SpaceWire. The National Aeronautics and Space Administration (NASA) at the Goddard Spaceflight Center in Greenbelt, Maryland, developed and specified a reliable protocol for its Geostationary Operational Environment Satellite known as GOES-R Reliable Data Delivery Protocol (GRDDP) [1].

The JAS program implemented and tested GRDDP and then suggested a number of modifications to the original specification to meet its program-specific requirements. This document details the full Reliable Data Delivery Protocol (RDDP) specification as modified for JAS.

1.1 Scope

The JAS Reliable Data Delivery Protocol (JRDDP) uses the lower-level SpaceWire data link layer to provide reliable packet delivery services to one or more higher-level host application processes. This document specifies the functional requirements for JRDDP but does not specify the interfaces to the lower- or higher-level processes, which may be implementation-dependent.

1.2 GRDDP Changes

Table 1 highlights the changes that were made to NASA's GRDDP specification and provides a short rationale for those changes.

Table 1. JRDDP Modifications.

Change	Rationale
Added variable-length source address	Accommodates larger and/or hierarchical network configurations
Increased Cyclic Redundancy Check (CRC) to 16 bits	Improves error detection capability for larger packet sizes
Added optional secondary header	Allows program-specific modifications and growth of the protocol with only minimal changes to the specification baseline
Urgent messaging is now optional	Reliable data delivery is the primary focus of JRDDP and best-effort delivery can be handled by other protocols
Added version number	Facilitates detection of unsupported protocol variants
Added sequence flags	Facilitates fragmentation and reassembly of data buffers passed to/from JRDDP
Added close channel semantics	Facilitates resource reuse in a dynamic communications environment
Increased channel number to 16 bits	Expands the number of defined channels

2. DEFINITIONS

Byte: An octet or 8 bits.

CONTROL Packet: Control packets cause a Transport End Point (TEP) to transition from one state to another. The OPEN/RESET and CLOSE packets constitute JRDDP CONTROL packets.

Maximum Transmission Unit (MTU): The size (in bytes) of the largest JRDDP packet (header, payload, and trailer) that the protocol can transmit. MTU size is implementation-dependent; however, the MTU size shall be greater than the size of the header and trailer but less than or equal to 65536 (64 KB) bytes.

Nibble: Four (4) bits.

Receiver: An electronic circuit that receives signals over a physical medium.

Secondary Header: An optional variable-length header that is located at the beginning of the payload field and that extends the information that can be contained in the packet header. The contents of the secondary header are implementation-dependent and shall be documented in a program-specific document.

SpaceWire Link: A bidirectional point-to-point connection between two SpaceWire ports.

SpaceWire Port: SpaceWire transmitter and receiver circuits and associated logic that implements the SpaceWire Exchange-level protocol including link initialization, character flow control, and link error detection and recovery.

Transmitter: An electronic circuit that transmits signals over a physical medium.

Transport Channel: A protocol-defined data path between two TEPs. A transport channel can exist only between one transmit TEP and one receive TEP. Each transport channel is a one-way data path for application packets. The protocol supports multiple concurrent transport channels over a SpaceWire Link.

Transport End Point (TEP): Defined on a host system for the purpose of either transmitting or receiving application packets over a SpaceWire Link. Multiple TEPs can be defined for any host system, but each TEP can only transmit or receive, not both.

3. OVERALL FUNCTIONAL DESCRIPTION

This protocol describes the mechanism for reliable transfer of data packets over a SpaceWire connection and adds the following capabilities to a SpaceWire Link:

- Multiplexed Logical Connections
- Reliable Delivery
- Missing packet detection
- Out-of-sequence packet reordering
- Butter fragmentation and reassembly

3.1 Multiplexed Logical Channels

The protocol shall support multiple simultaneous logical connections over a single SpaceWire Link.

3.1.1 Channel Independence

Each transport channel shall operate independently from other transport channels.

3.1.2 Transmit Priority

When more than one packet is available for transmit, all acknowledgment (ACK) packets shall be transmitted first, then CONTROL packets, then URGENT packets (if implemented), then Retransmit packets (CONTROL or DATA), then DATA packets.

3.1.3 Data Transmit Queue

When packets from more than one channel are available for transmit, packets shall be transmitted in the order in which they are queued and in accordance with the transmit priority described above.

3.1.4 Urgent Messaging (Optional)

Urgent messaging is a best-effort priority communications path that does NOT provide for reliable delivery, missing packet detection, or out-of-sequence packet reordering. It may optionally be implemented in this protocol to facilitate an out-of-band transport pathway that may be useful for such things as time broadcasts, exception/error control, meta messages, etc.

3.2 Reliable Delivery

JRDDP detects lost packets, duplicate packets, out-of-sequence packets, and provides damaged data recovery. The protocol provides additional error detection beyond the SpaceWire physical layer by utilizing Cyclic Redundancy Checks (CRCs), packet sequence numbers, positive acknowledgment, and timeouts to detect lost or duplicated packets.

3.2.1 Error Detection

Packet errors shall be detected by adding a CRC to each packet transmitted, checking it at the receiver, and discarding any erroneous packet.

3.2.2 Packet Sequence Numbers

An 8-bit sequence number shall be assigned to each packet transmitted.

3.2.3 Sequence Number Use

At the receiver TEP the sequence numbers shall be used to detect lost packets and duplicate packets, and to correctly order packets.

3.2.4 Acknowledgment and Retransmit

The receiver shall send a positive ACK packet for each DATA packet received without error.

3.2.5 Retransmission

If an ACK packet is not received by the transmit TEP within a defined **channel-specific** timeout interval, the DATA packet shall be retransmitted.

3.3 Buffer Fragmentation and Reassembly

In cases where a higher-level transmit application (or higher layer in a communications stack) must transmit a buffer of data that exceeds the JRDDP MTU size, the buffer must be fragmented into a series of JRDDP packets and then reassembled at the receive TEP before returning the buffer to the next higher level.

Sequence numbers in the JRDDP header assure the correct ordering of individual JRDDP packets within the series of packets that represent the buffer and sequence flags delineate the boundaries of the data buffer within a stream of packets.

A maximum buffer size must be agreed upon by both transmit and receive TEPs to preclude excessive resource allocations.

4. PACKET FORMAT

All JRDDP packets shall include a variable-length header, followed by a variable-length payload, followed by a 2-byte CRC Trailer. Figure 1 shows how the JRDDP packet is embedded within the standard SpaceWire packet. Note that while a SpaceWire packet may have zero or more destination addresses before the JRDDP header, the JRDDP requires that exactly one destination address be delivered to the protocol logic.

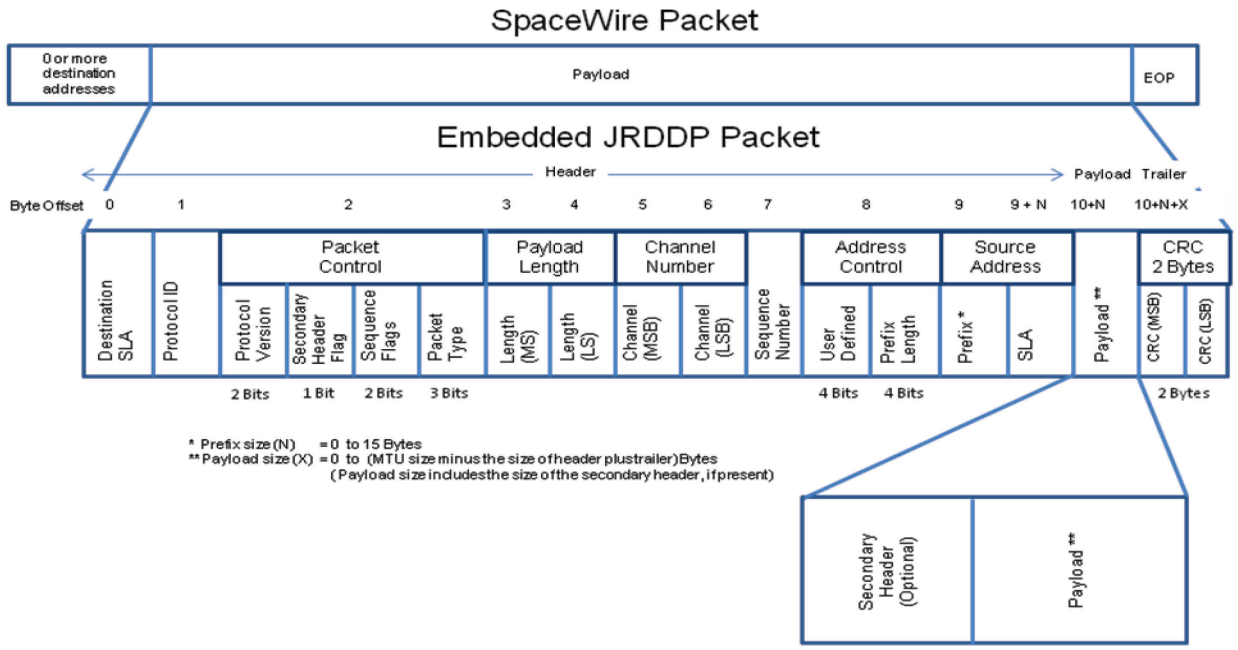

Figure 1. JRDDP packet format.

4.1 Header

4.1.1 Destination Address

The first byte of the header shall contain the Destination SpaceWire Logical Address (SLA) field that is associated with the destination TEP to which the packet is being sent.

4.1.2 Protocol ID

The second byte of the header shall contain the SpaceWire Protocol ID field and is assigned a value of decimal 238 (TBD) by the European Cooperation for Space Standardization.

4.1.3 Packet Control

The third byte of the header shall contain the Packet Control field.

4.1.3.1 Protocol Version Number

The two most significant bits of the Packet Control field shall contain the protocol version number of the JRDDP protocol specification to which the format of the packet complies. These bits shall be set to 0x1 for this version of the specification.

4.1.3.2 Secondary Header Flag

A 0x1 in bit 5 of the Packet Control field shall indicate the presence of a variable length secondary header in the Payload field of the packet. A 0x0 indicates the absence of the secondary header.

4.1.3.3 Sequence Flags

A two-bit field of Sequence Flags (bits 3-4) of the Packet Control field is used to signal the first, intermediate, and final packets in a series of JRDDP packets that collectively represent a data buffer. The Sequence Flags are set by the transmit TEP during fragmentation and are used by the receive TEP for data buffer reassembly. Values for the Sequence Flags are shown in Table 2.

Table 2. Buffer Sequence Flag Values.

Sequence Flags	Value
First packet of the buffer	1
Continuation packet	0
Last packet of the buffer	2
Stand-alone packet	3

CONTROL and ACK packets shall have their Sequence Flags bits set to 0x3 indicating that they are stand-alone packets.

4.1.3.4 Packet Type

The three least significant bits (0-2) of the Packet Control field shall identify the type of packet. Packet-type values are listed in Table 3.

Table 3. Packet-Type Values.

Packet Type	Value
Application Data	0
Acknowledge	1
Open/Reset Command	2
Close Command	3
Urgent	4
Reserved	5-7

4.1.4 Payload Length

The Payload Length field in bytes 4 and 5 of the header specifies the number of bytes in the Payload field and is a 16-bit value. It does not include any bytes in the JRDDP header/trailer but DOES include bytes used in the secondary header.

The fourth header byte shall contain the most significant byte of the Payload Length field and the fifth header byte shall contain the least significant byte of the Payload Length field.

The maximum size of the Payload field shall be constrained to be the MTU size minus the maximum length of the header plus trailer fields.

4.1.5 Channel Number

Channel Numbers are 16-bit values that uniquely identify multiplexed logical connections between two TEPs. Channel Number values are implementation-dependent and are generally assigned at a level above the JRDDP.

The sixth byte of the header shall contain the most significant byte of the Channel Number and the seventh byte of the header shall contain the least significant byte of the Channel Number.

Note: One purpose of a secondary header may be to augment the size of the Channel Number field in a dynamic communications environment.

4.1.6 Sequence Number

The eighth byte of the header shall contain the Sequence Number field and be a value in the range of 0 through 255.

4.1.7 Address Control

The ninth byte of the header shall contain the Address Control field.

4.1.7.1 User Defined

The most significant nibble of the Address Control field is user-defined and shall be filled with zeros unless a program using this protocol defines them in a program-specific document.

4.1.7.2 Prefix Length

The four bits of the least significant nibble of the Address Control field are used to specify the number of addressing bytes that are to be prepended to the Source Address SLA to form a full variable-length address. Valid values are 0 to 15.

4.1.8 Source Address

The JRDDP implements a variable-length source address whose purpose is to accommodate those systems that may require multiple address bytes to uniquely identify an endpoint, such as in SpaceWire path addressing or regional addressing.

The Source Address of the packet shall consist of a variable-length number of addressing bytes. The combination of the Prefix and SLA fields constitutes the fully qualified node address from which the packet was sent. The number of bytes contained within the Prefix field is specified in the Prefix Length bits of the Address Control field.

4.1.8.1 Prefix

The Source Address Prefix bytes contain all of the variable-length addressing bytes except for the SLA. The Prefix is not used when the Address Control Prefix Length is zero (0x0).

4.1.8.2 SLA

The SLA byte of the Source Address Field shall be the SLA that identifies the node from which the packet was sent. In the event of a variable-length address, the SLA is the last byte of the fully qualified Source Address.

4.2 Payload

4.2.1 Secondary Header (Optional)

The presence of the optional secondary header is indicated by the Secondary Header Flag in the Packet Control field. If a secondary header is present, its contents and format are implementation-dependent and shall be documented in a program-specific document. The secondary header occupies the first bytes of the Payload field.

4.2.2 DATA Packets and URGENTs

The protocol's DATA packets (and URGENTs, if implemented) shall contain a Payload field containing from 0 to X bytes of content for delivery to the receive TEP. The maximum value for X is the MTU size minus the sizes of the header and the trailer. If a secondary header is present, the maximum size of the Payload field must be reduced by the size of the secondary header.

4.2.3 ACK and CONTROL Packets

The Payload field shall be unused (zero bytes) for ACK and CONTROL packets unless a secondary header is present.

4.3 Trailer

The JRDDP packet trailer shall be a 16-bit CCITT CRC computed from the Destination SLA to the last byte of the Payload field.

4.3.1 CRC

The 16-bit CCITT CRC shall be computed according to the following polynomial:

$$x^{16} + x^{12} + x^{5} + 1$$

Before computing each packet's CRC, the initial value for the computation shall be set to all 1s.

5. CHANNEL OPERATIONS

5.1 TEP Parameters

Each TEP shall be defined with the parameters shown in Table 4.

Table 4. TEP Parameters.

Local SLA	The SLA assigned to the TEP for each channel.
Remote SLA	The SLA assigned to the TEP that is connected to the local TEP.
TEP Type	Identifies the TEP as transmit or receive.
Window Size	The size of the channel's sliding window. The window size must be a power of 2.
Time Out	Transmit TEPs only. The time to wait to for an acknowledgment before retransmitting a DATA or CONTROL packet.
Maximum Retries	Transmit TEPs only. The number of retry attempts allowed before declaring a channel failure and initiating a channel reset.
Close Time Out	Receive TEPs only. The time to wait after entering the CLOSING state before transitioning to the CLOSED state.

5.2 TEP States

Each TEP shall be in one of four possible operating states shown in Table 5.

Table 5. TEP States.

CLOSED	The TEP does not generate any packets on the link and does not respond to any packets received.
ENABLED	A TEP transitions to the ENABLED state when the host has requested it to be opened and provided input/output buffer resources. In addition, a Transmit TEP sends an OPEN/RESET packet on this transition.
OPEN	A receive TEP transitions from ENABLED to OPEN when an OPEN/RESET packet has been received from the transmit TEP. A transmit TEP transitions from ENABLED to OPEN when an OPEN/RESET packet is acknowledged by the receive TEP.
CLOSING	A transmit TEP transitions to the CLOSING state from either the ENABLED or OPEN states when commanded by the host. In addition, the transmit TEP sends a CLOSE packet on this transition but does not transmit any other packets. The transmit TEP stays in the CLOSING state until it receives an ACK from the receive TEP or it exhausts its retry count. When either of these conditions is satisfied the transmit TEP transitions to the CLOSED state. A receive TEP transitions to the CLOSING state when a CLOSE packet is received and remains in this state until the expiration of the close timeout occurs. While in the CLOSING state the receive TEP acknowledges all CLOSE packets. After the expiration of the timeout, the receive TEP transitions to the CLOSED state.

5.2.1 Channel Closing Process

The following sections present the interactions (in the form of Unified Modeling Language (UML) Sequence Diagrams) between transmit and receive endpoints while closing a channel. Various error conditions are presented in order to show nominal, delayed, and worst-case scenarios.

The hexagonal icons represent the TEP states while the dashed green lines represent data flows (packets) between the endpoints across a link. In the case where errors are present and a packet is either corrupted or lost, the UML lost message icon is used to denote this error condition instead of the dashed green line.

5.2.1.1 Nominal Case

In the nominal case (Figure 2), the CLOSE packet is immediately acknowledged by the receive TEP and the transmit TEP transitions to the CLOSED state. The receive TEP transitions to the CLOSED state after the expiration of its timer. The purpose of the receive TEP's delay in the CLOSING state is to remain responsive to CLOSE packets that may be retransmitted by the transmit TEP if the ACK packet is lost or corrupted.

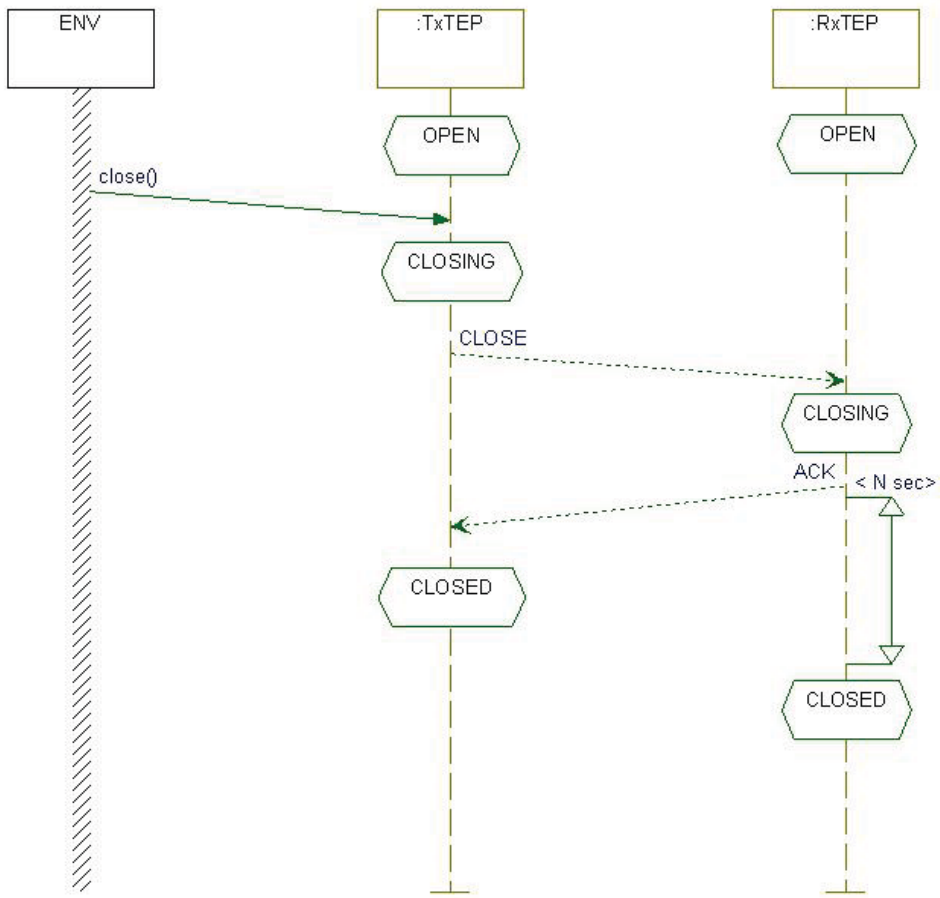

Figure 2. Nominal case.

5.2.1.2 Delayed Case

A delayed but recoverable scenario occurs when a receive TEP receives at least one CLOSE packet but the transmit TEP does not receive any ACK packets, as shown in Figure 3. In this case the receive TEP will eventually transition to the CLOSED state via the timeout and the transmit TEP will eventually transition to the CLOSED state when its retry count is exhausted.

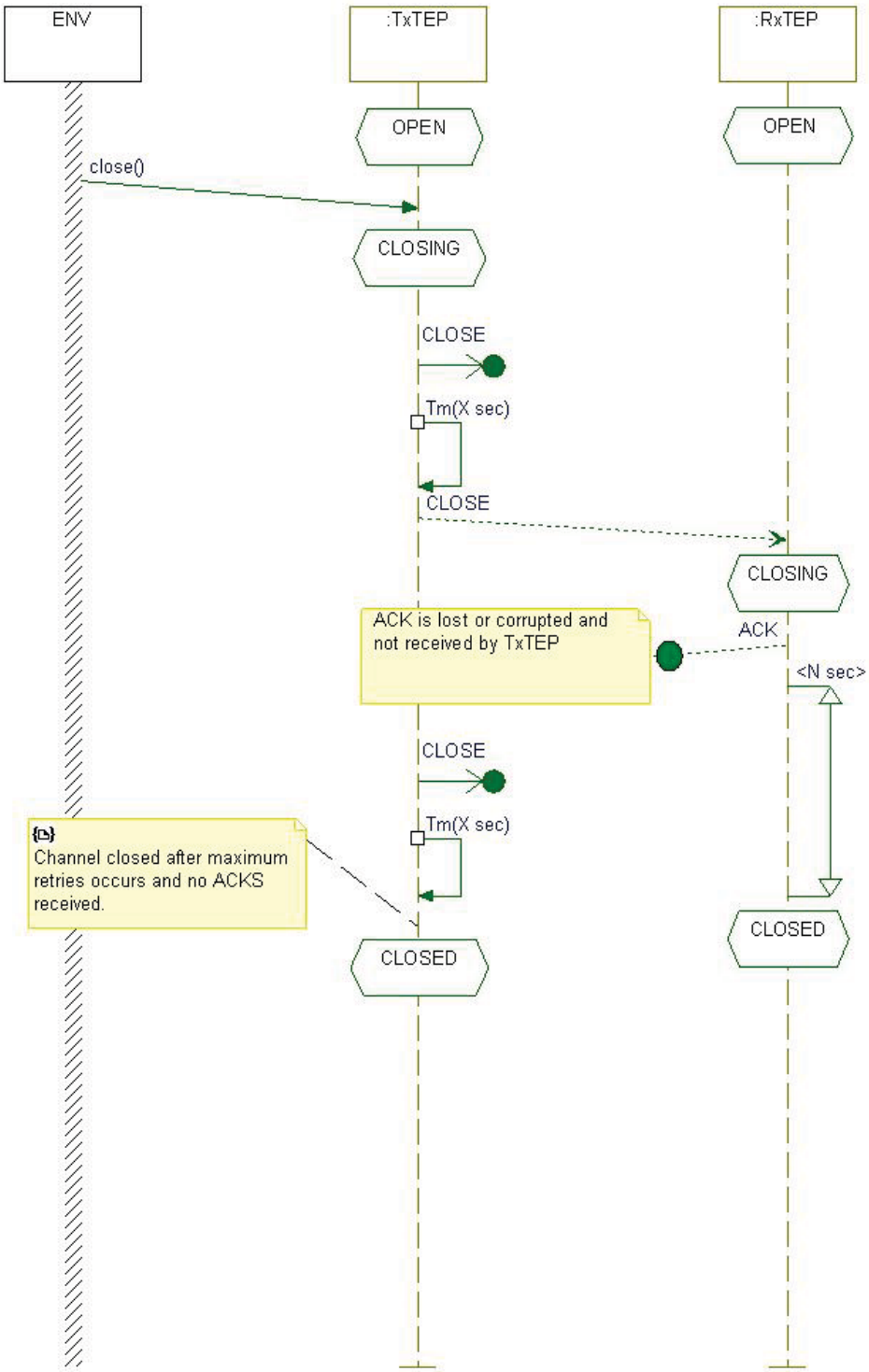

Figure 3. Delayed case.

5.2.1.3 Worst Case

If the link between the transmit and receive TEPs is bad, i.e., packets are corrupted or lost, and the receive TEP does not receive at least one CLOSE packet, the transmit TEP will transition to the CLOSED state after all retries have been exhausted and the receive TEP will remain in the OPEN state, as shown in Figure 4.

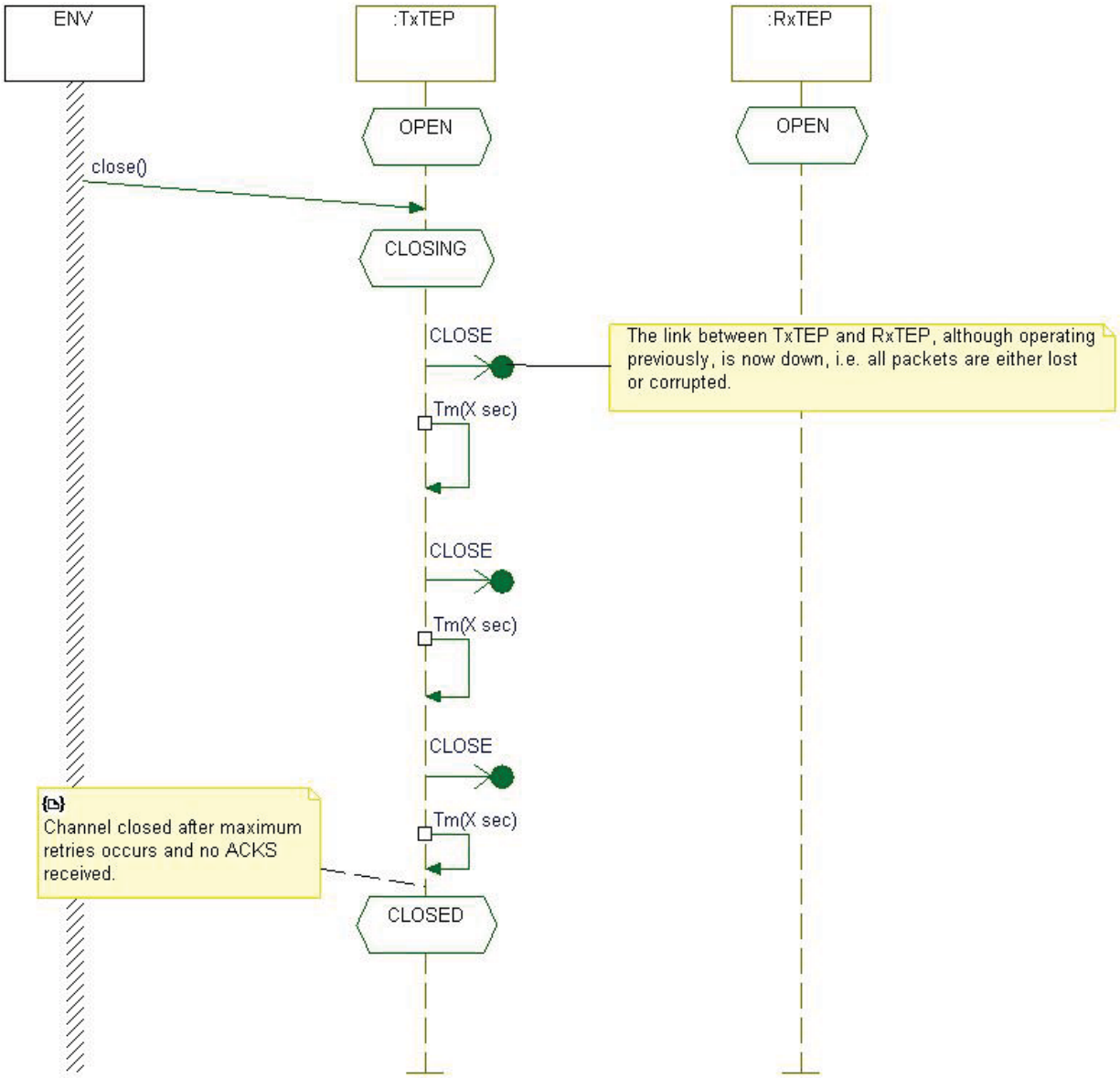

Figure 4. Worst case.

This represents the worst-case scenario and results in inconsistent states between transmit and receive TEPs. The only recovery from this inconsistency is to reopen the transmit TEP's channel or perform a power down and restart. If the channel is sufficiently bad to create this condition, the power down/restart may be the only recourse in restoring the link.

Because of the "one-way" nature of the data flows in JRDDP channels there is no way to guarantee that both the receive TEP and transmit TEP know when the channel is closed. Timeouts and retries provide the most reasonable mechanism for managing the channel closing process.

5.3 Logical Connections

Upon initialization all TEPs shall be in the CLOSED state.

5.3.1 OPEN/RESET Command

When a transmit TEP transitions to the ENABLED state (as commanded by the host), it shall send an OPEN/RESET packet to its receive TEP and initiate an acknowledgment timer.

5.3.2 CLOSE Command

When a transmit TEP transitions to the CLOSING state (as commanded by the host), it shall send a CLOSE packet to its remote receive TEP and initiate an acknowledgment timer.

5.3.3 CONTROL Timer Cancellation

Upon receipt of a CONTROL packet acknowledgment, a transmit TEP shall cancel its acknowledgment timer for that CONTROL packet.

5.3.4 CONTROL Timer Expiration

Upon expiration of the retransmit timeout period, the transmit TEP shall retransmit the CONTROL packet.

5.4 Transport Channel Connection

A transport channel connection shall be considered established when a transmit TEP and receive TEP are both in the OPEN state.

5.5 Receive TEP Operations

A receive TEP shall not send DATA or CONTROL packets. It may only send ACK packets.

5.5.1 Sliding Window

The receive TEP shall maintain a sliding window, which is a range of consecutive sequence numbers that is used to determine whether each received DATA packet will be accepted or discarded. The size of the sliding window shall be as agreed between the transmit and receive TEPs.

5.5.2 Sliding Window Size

The sliding window size must be a power of 2 and the maximum sliding window size shall be 256.

5.5.3 Sliding Window Range

The receive window range shall start with the sequence number of the next DATA packet expected to be delivered and end with sequence number equal to the start plus Window Size minus 1.

5.5.4 Window Advance

The receive window shall be advanced by 1 upon receipt of a packet containing the next expected sequence number.

Note: If packets with successively adjacent sequence numbers have already been received out of order, the start of the receive window will be advanced by more than 1, plus the number of successively adjacent "early" packets.

5.5.5 Packet Acknowledgment

All DATA and CONTROL packets received without error shall be acknowledged except as indicated in Section 5.5.7.

5.5.6 Packets with Errors

Any packet received with detectable errors shall be discarded and not acknowledged

5.5.7 Out of Window Sequence Number

A DATA packet that is received with a sequence number that is not within the receive window shall **NOT** be acknowledged, but discarded.

5.5.8 Duplicate Sequence Number

A DATA packet received with a sequence number within the receive window that is a duplicate of a packet pending delivery to the host shall be acknowledged, but discarded.

5.5.9 URGENT Acknowledgment (If Implemented)

URGENT packets shall not be acknowledged.

5.5.10 URGENT Delivery Order (If Implemented)

URGENT packets shall be delivered to the host in the order received.

5.5.11 URGENT Delivery Priority (If Implemented)

URGENT packets shall be delivered to the host before any DATA packets pending delivery.

5.5.12 CONTROL Packet Sequence Number

A CONTROL packet that does not have a sequence number of zero shall be treated as an error packet and discarded.

5.5.13 OPEN/RESET Command Processing

When an OPEN/RESET packet is received, the receive window start shall be set to 1.

5.5.14 CLOSE Command Processing

When a CLOSE packet is received the packet shall be acknowledged and the TEP shall transition to the CLOSING state.

5.5.15 Packets Pending Delivery

When a CONTROL packet is received all packets pending delivery to the host shall be discarded. Buffer fragmentation and reassembly operations, if active, shall be reset.

5.5.16 OPEN/RESET Command Report

Receipt of an OPEN/RESET packet after a channel is opened indicates an error on the channel and shall be reported to the next higher level, i.e., communications layer or application. The error report shall include the channel number that was reset.

5.5.17 CLOSE Command Report

Receipt of a CLOSE packet after a channel is opened shall be reported to the next higher level, i.e., communications layer or application. The report shall include the channel number that was closed.

5.6 Transmit TEP Operations

5.6.1 Transmit TEP ACKs

A transmit TEP shall not send an ACK packet.

5.6.2 Transmit TEP Sequence Number Allocation

Each DATA packet transmitted shall have a sequence number allocated from the TEP's transmit window range of available sequence numbers.

5.6.3 CONTROL Packet Sequence Number

All CONTROL packets shall be transmitted with a sequence number of zero.

5.6.4 Transmit Window

A transmit TEP shall maintain a sliding window range of consecutive sequence numbers that are available for transmitting DATA packets.

5.6.5 Unacknowledged Packets

The transmit window shall limit the number of unacknowledged DATA packets that can be transmitted and prevents transmit operations of DATA packets outside the sliding window until the ACK packet corresponding to the first DATA packet in the window has been received. This process effectively throttles transmit operations.

5.6.6 Transmit Window Start

The transmit window start shall be set to 1 when a CONTROL packet's ACK is received.

5.6.7 Transmit Window Advance

The transmit window start **shall** be advanced by 1 when the ACK is received for the first sequence number in the transmit window.

5.6.8 Packet Retransmit

A transmitted DATA packet that is not acknowledged within a channel-specific timeout interval shall be retransmitted with the original sequence number up to a channel-specific number of times.

5.6.9 Retry CONTROL

When a channel-specific number of retry attempts have been exceeded, the channel shall be reset.

5.6.10 Timeout Start

The timeout interval shall begin when the last byte of the DATA or CONTROL packet has been transmitted.

5.6.11 URGENT Packet Transmission (If Implemented)

URGENT packets shall be sent immediately without being allocated a transmit window sequence number or starting an acknowledgment timer.

Note: URGENT packets are sent once without retires or acknowledgments.

6. REFERENCES

[1] *GOES-R Reliable Data Delivery Protocol (GRDDP)*, 417-R-RPT-0050, RM Version, NASA Goddard Space Flight Center, Greenbelt, Maryland, January 16, 2008.

[2] *European Cooperation For Space Standardization SpaceWire-Links, Nodes, Routers and Networks*, ECSS-E-50-12A, January 24, 2003.

APPENDIX A. JOINT ARCHITECTURE STANDARD (JAS) RELIABLE DATA DELIVERY PROTOCOL (RDDP) SPECIFICATION PROGRAM SUPPLEMENT

A.1 Introduction

The Joint Architecture Standard (JAS) program is a specification for hardware and software that can be used as the building blocks for space-based payloads. The JAS Reliable Data Delivery Protocol (JRDDP) provides a reliable data delivery mechanism for data transfer between JAS nodes. The JRDDP provides for the embedding of program specific information in the JRDDP's secondary headers.

This document defines the secondary headers used by the JAS program to extend JRDDP and provides implementation details that are unique to the JAS program.

A.2 Definitions

See JAS-JRDDP-00001, Version D.

A.2.1 Standard Secondary Header

The Standard Secondary Header is the first of two Secondary Header formats used by the JAS program. It purpose is to increase the size of the channel number used in JRDDP from 16 to 32 bits.

A.2.2 Augmented Secondary Header

The Augmented Secondary Header is the second Secondary Header format used by the JAS program and is only used during the channel open/reset process. Its purpose is to facilitate the negotiation of sliding window sizes used by transmit and receive Transport End Points (TEPs).

A.3 Packet Format

Figure A-1 is an excerpt from the JRDDP specification and shows the placement of the optional Secondary Header within a JRDDP packet.

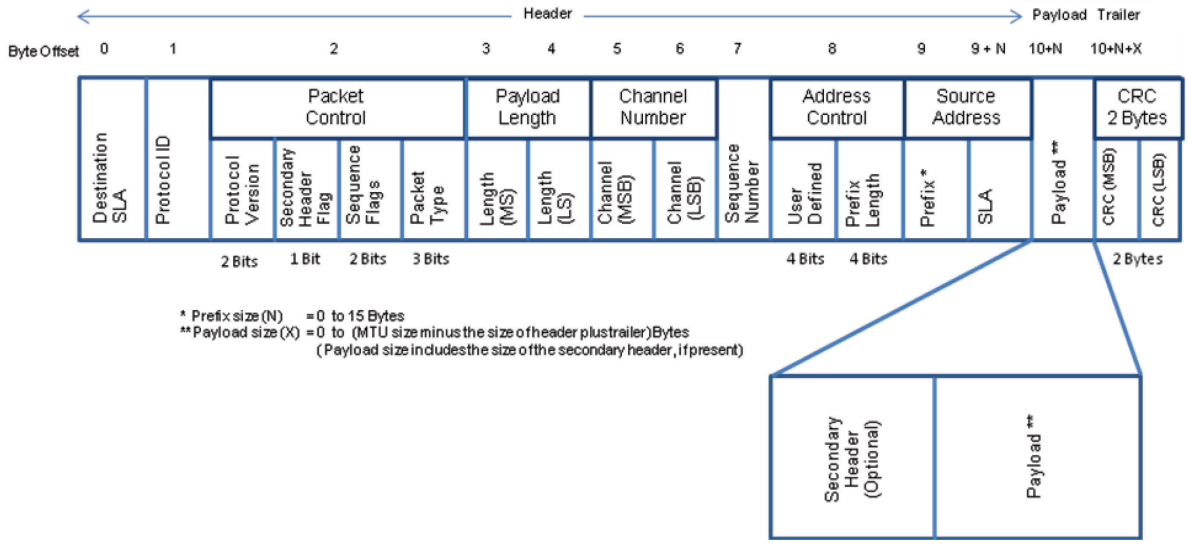

Figure A-1. JRDDP packet format.

The JAS program **shall** use the JRDDP's secondary headers to extend the JRDDP and supply additional information, as defined in the following sections.

A.4 JAS Program JRDDP Secondary Headers

JAS Secondary Headers consist of at most 4 bytes, as shown in Figure A-2.

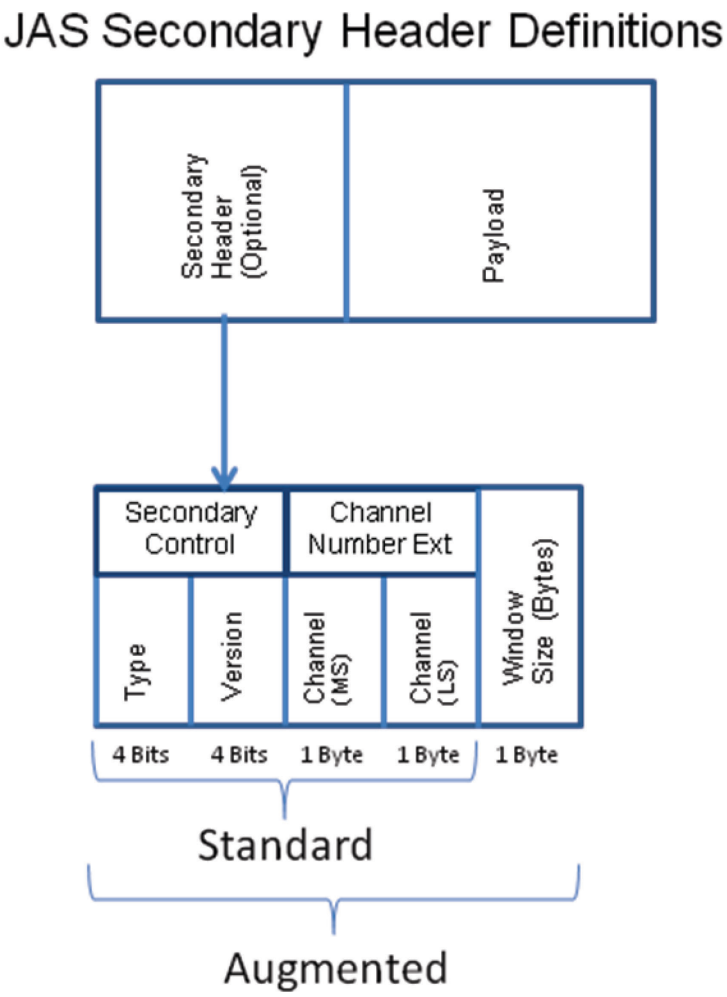

Figure A-2. JAS Secondary Header format.

Definitions for each of the Secondary Header fields follow.

A.4.1 Secondary Control

The most significant nibble of the Secondary Control field **shall** contain the Secondary Header type. The values for this field are shown in Table A-1.

Table A-1. Secondary Header Type Values.

Secondary Header Type	Value
Standard	0
Augmented	1
Unused	2 to15

A.4.2 Version

The least significant nibble shall contain the version number of the Secondary Header. The value for this nibble shall be 0x0 for this version of JRDDP-PSUP.

A.4.3 Channel Number Extension

Bytes two and three of the Secondary Header **shall** contain an extension for the primary header's Channel Number field. The most significant byte of the Channel Number Extension shall be byte two of the Secondary Header and the least significant byte of the Channel Number Extension **shall** be byte three of the Secondary Header.

A.4.4 Window Size (Augmented Header Only)

The fourth byte of the Secondary Header is only used with OPEN/RESET packets and their corresponding acknowledgment (ACK) packets and contains the size of the sliding window in use by the TEP.

A.5 Implementation Details

A.5.1 Urgent Messaging

JRDDP is but one of several communication pathways used on JAS platforms; therefore, urgent messaging, as defined in JRDDP, **shall not** used for JAS.

A.5.2 Secondary Headers

The JAS program **shall** use a Secondary Header in each packet transmitted by a TEP. Standard Secondary Headers shall be used in all but the OPEN/RESET packets and their ACK packets. Augmented Secondary Headers shall be used for the OPEN/RESET packets and their ACKs.

A.5.3 Extended Channel Numbers

The Channel Number Extension field **shall** be appended to the 16-bit Channel Number contained in the JRDDP header in order to create a unique 32-bit channel number used by both transmit and receive TEPs.

A.5.4 Sliding Window Size Negotiations

A transmit TEP inserts the desired size of its sliding window into the Window Size field of the Augmented Secondary Header in its OPEN/RESET packet.

Upon receipt of the OPEN/RESET packet, a receive TEP should configure itself to use the window size that is found in the Augmented Secondary Header, if possible. If this is not possible, it should configure itself to use a sliding window size that is less than or equal to the window size provided by the transmit TEP and greater than or equal to 1. The receive TEP then inserts its sliding window size into the Window Size field of the Augmented Secondary Header for the ACK packet that is returned.

Upon receiving an ACK for its OPEN/RESET packet, the transmit TEP **shall** adjust its sliding window size to the value provided in the Window Size field of the Augmented Secondary Header in the ACK packet from the receive TEP.

A.5.5 Secondary Header Verification

Transmit and receive TEPs **shall** verify that the version number contained in the JRDDP packet's Secondary Header is compatible with the JRDDP packet definitions supported by the JRDDP implementation.

A.6 References

[1] *Joint Architecture System Reliable Data Delivery Protocol (JRDDP)*, JRDDP-00001, Version B, Sandia National Laboratories, Albuquerque, NM, November 23, 2010.

DISTRIBUTION

1	MS0501	Neill P. Symons	5337
1	MS0503	James W. Daniels	5337
1	MS0503	Dominic A. Perea	5337
1	MS0503	Mythi M. To	5337
1	MS0503	Christopher K. Wojahn	5337
1	MS0513	Kevin L. Harrison	5336
1	MS0513	Richard D. Hunt	5336
1	MS0521	Ed E. Boucheron	2617
1	MS0532	Michael Gardner	5348
1	MS0533	Kurt W. Sorensen	5345
1	MS0621	Dallas Wiener	5632
1	MS0661	Daniel E. Gallegos	2623
1	MS0661	Aaron D. Niese	2623
1	MS0672	Brian P. Van Leeuwen	5628
1	MS0966	Veronica Chavez-Soto	5733
1	MS0971	Ethan L. Blansett	5735
1	MS0980	Matt P. Napier	5571
1	MS0982	Jaime Gomez	5732
1	MS0982	Dan J. Kral	5732
1	MS0986	David M. Bullington	2664
1	MS0986	Jonathon W. Donaldson	2664
1	MS0986	Justin Wayne Enderle	2664
1	MS0986	David Heine	2664
1	MS0986	Jeffrey L. Kalb	2664
1	MS0986	David S. Lee	2664
1	MS0986	J. (Heidi) Ruffner	2664
1	MS1027	John M. Eldridge	5632
1	MS1243	Raymond H. Byrne	5535
1	MS0899	Technical Library	9536 *(electronic copy)*

Sandia National Laboratories