



# SpaceWire-R

2023年3月31日 制定

宇宙航空研究開発機構

#### 免責条項

ここに含まれる情報は、一般的な情報提供のみを目的としています。JAXA は、かかる情報の正確性、有用性又は適時性を含め、明示又は黙示に何ら保証するものではありません。また、JAXA は、かかる情報の利用に関連する損害について、何ら責任を負いません。

#### Disclaimer

The information contained herein is for general informational purposes only. JAXA makes no warranty, express or implied, including as to the accuracy, usefulness or timeliness of any information herein. JAXA will not be liable for any losses relating to the use of the information.

#### 発行

〒305-8505 茨城県つくば市千現 2-1-1

宇宙航空研究開発機構 安全・信頼性推進部

JAXA (Japan Aerospace Exploration Agency)

本書は英語で書かれた草案を日本語に翻訳し、日本の宇宙機関 JAXA により制定された。  
本標準は日本語を正とする。ただし、図表の一部で英語表記しかないものについては、それらが正本となる。文章の内容に疑問点がある場合は、日本語及び英語の双方を参照の上、JAXA 安全・信頼性推進部まで連絡をすること。

This document was originally drafted in English, then subsequently translated into Japanese and authorized by the Japanese space agency, JAXA.

The English translation is for reference purposes only, except for some tables and figures that contain English only, in which case they are the original. If there is anything ambiguous about the content of the text, please refer to both the Japanese version and the English version and contact JAXA Safety and Mission Assurance Department.

# SpaceWire-R

SCDHA 151-1.00

Issue 1.00

Japan Aerospace Exploration Agency (JAXA)

This document was originally drafted in English, then subsequently translated into Japanese and authorized by the Japanese space agency, JAXA.

The English translation is for reference purposes only, except for some tables and figures that contain English only, in which case they are the original. If there is anything ambiguous about the content of the text, please refer to both the Japanese version and the English version and contact JAXA Safety and Mission Assurance Department.

本書は英語で書かれた草案を日本語に翻訳し、日本の宇宙機関 JAXA により制定された。

本標準は日本語を正とする。ただし、図表の一部で英語表記しかないものについては、それらが正本となる。文章の内容に疑問点がある場合は、日本語及び英語の双方を参照の上、JAXA 安全・信頼性推進部まで連絡をすること。

# CONTENTS

<b>SPACEWIRE-R.....</b>	<b>1</b>
1 INTRODUCTION // はじめに.....	1
1.1. Purpose // 目的 .....	1
1.2. Scope // 範囲.....	1
1.3. Applicability // 適用先 .....	1
1.4. References // 関連文書 .....	2
1.5. Document Structure // 文書の構造.....	3
1.6. Definitions and Notations // 定義及び表記法.....	4
1.7. Verbal Forms // 表現形式 .....	11
2. OVERVIEW // 概要.....	13
2.1. Overall Protocol Architecture 全体的なプロトコルアーキテクチャ .....	14
2.2. Protocol Features // プロトコル機能 .....	15
2.3. Relationship with other Protocols // 他のプロトコルとの関係 .....	19
3. SERVICE DEFINITION // SERVICE の定義 .....	21
3.1. General // 一般.....	21
3.2. Channel Control Service.....	21
3.3. Data Transfer Service // データ伝送 Service .....	25
4. PROTOCOL SPECIFICATION // プロトコル仕様.....	31
4.1. General // 一般.....	31
4.2. SPW-R Packets.....	32
4.3. Structure of Procedures // Procedures の構造.....	51
4.4. Transmit TEP Procedures // 送信 TEP Procedures .....	53
4.5. Receive TEP Procedures // 受信 TEP Procedures.....	73
4.6. Node Procedures .....	95
5. HOW TO SPECIFY PARAMETERS FOR EACH NETWORK 各 NETWORK のパラメータを定める方法 99	
5.1. General // 一般.....	99
5.2. Parameters Used for each SpaceWire Network 各 SpaceWiRe NETWork に用いるパラメータ .....	99
5.3. Parameters Used for each Transport Channel 各 Transport Channel に用いるパラメータ 99	
APPENDIX A. ACRONYMS.....	101

APPENDIX B. EXAMPLES OF PDU EXCHANGES // PDU 交換の例 .....	103
B.1. General // 一般.....	103
B.2. Normal Case // 通常のケース .....	103
B.3. Data Error Case // データエラーの場合 .....	104
B.4. ACK Error Case // ACK エラーケース .....	105
B.5. Flow Control Case // Flow Control の場合 .....	106
APPENDIX C. EXAMPLE OF PARAMETERS FOR A NETWORK // NETWORK のためのパラメータ例	
107	
C.1. General // 一般.....	107
C.2. Parameters Used for a SpaceWire Network ある SpaceWire Network に用いるパラメータ..	107
C. 3. Parameters Used for a Transport Channel ある Transport Channel に用いるパラメータ....	108

# 1 INTRODUCTION // はじめに

## 1.1. PURPOSE // 目的

The purpose of this document is to specify a communications protocol called “[SpaceWire-R](#)” that provides onboard applications with reliable data transfer [services](#) over [SpaceWire \(SpW\)](#) [A1] [networks](#).

この文書の目的は、[SpaceWire \(SpW\)](#) [A1] [networks](#) を介した信頼性の高いデータ伝送 [services](#) を、衛星搭載アプリケーションに提供する通信プロトコル“[SpaceWire-R](#)” の仕様を定める事である。

## 1.2. SCOPE // 範囲

This document specifies [SpaceWire-R](#) in terms of the interfaces between [SpaceWire nodes](#). It does not specify how these interfaces should be implemented with hardware or software.

この文書は、[SpaceWire nodes](#) 間のインタフェースの観点から [SpaceWire-R](#) を定める。これらのインタフェースをハードウェアやソフトウェアで如何に実装すべきかは定めない。

## 1.3. APPLICABILITY // 適用先

This document applies to interfaces of subsystems and instruments onboard spacecraft and their ground support equipment all of which conform to [SpaceWire-R](#).

この文書は、[SpaceWire-R](#) に準拠する、衛星搭載のサブシステムと機器、並びに、それらの地上支援装置のインタフェースに適用する。



## **1.4. REFERENCES // 関連文書**

### **1.4.1. NORMATIVE REFERENCES // 引用文書**

[A1] European Cooperation for Space Standardization (ECSS), “Space Engineering - SpaceWire - Links, nodes, routers and networks,” ECSS-E-50-12A, January 2003.

[A2] International Standard Organization (ISO), “Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model,” International Standard, ISO/IEC 7498-1, 2nd ed., 1994.

[A3] International Standard Organization (ISO), “Information Technology - Open Systems Interconnection - Basic Reference Model - Conventions for the definition of OSI services,” International Standard, ISO/IEC 10731, 1994.

[A4] European Cooperation for Space Standardization (ECSS), “Space Engineering - SpaceWire protocol identification,” ECSS-E-ST-50-51C, February 2010.

### **1.4.2. INFORMATIVE REFERENCES // 参考文書**

[R1] Sandia National Laboratories, “Joint Architecture Standard (JAS) Reliable Data Delivery Protocol (RDDP) Specification”, Sandia Report SAND2011-3500, May 2011.

[R2] Goddard Space Flight Center, “GOES-R Reliable Data Delivery Protocol (GRDDP)”, 417-R-RPT-0050, January 2008.

[R3] CCSDS, “Space Packet Protocol”, CCSDS 133.0-B-1, September 2003.

## 1.5. DOCUMENT STRUCTURE // 文書の構造

This document is organized as follows:

本書は次の通り構成する。

- a) Section 1 (this section) shows the purpose, scope, and applicability of the document, and lists the references, definitions, and notations used throughout the document,
  - b) Section 2 presents an overview of **SpaceWire-R**,
  - c) Section 3 defines the **services** provided by the **protocol entity**,
  - d) Section 4 specifies the **protocol data units** and **procedures** both of which are employed by the **protocol entity**,
  - e) Section 5 presents a standard method for specifying the parameters of **SpaceWire-R** used for a specific **SpaceWire network**,
  - f) Appendix A lists all acronyms used in this document,
  - g) Appendix B shows some examples of exchanges of **protocol data units (SpaceWire-R Packets)** between a sender and a receiver, and
  - h) Appendix C shows an example of specifying the parameters of **SpaceWire-R** used for a specific **SpaceWire network**.
- a) 1 章 (本章) は、本書の目的、範囲及び適用先を述べると共に、本書で用いる関連文書、定義、及び表記法を示す。
  - b) 2 章は、**SpaceWire-R** を概説する。
  - c) 3 章は、**プロトコル構成要素**が提供する **services** を定義する。
  - d) 4 章は、**プロトコル構成要素**が使う、**protocol data units** と **procedures** を定める。
  - e) 5 章は、具体的な **SpaceWire network** に用いる **SpaceWire-R** のパラメータを定める標準的な方法を示す。
  - f) Appendix A は、本書で用いる全略語を示す。
  - g) Appendix B は、送信者と受信者の間の **protocol data units (SpaceWire-R Packets)** の交換の例を幾つか示す。
  - h) Appendix C は、具体的な **SpaceWire network** に用いる **SpaceWire-R** のパラメータを定める例を示す。

## 1.6. DEFINITIONS AND NOTATIONS // 定義及び表記法

### 1.6.1. TERMS DEFINED IN THE SPACEWIRE // SPACEWIRE が定義する用語

This document adopts the following terms defined in “SpaceWire - Links, nodes, routers and networks” [A1]: 本書は、“SpaceWire - Links, nodes, routers and networks” [A1] が定義する次の用語を採用する。

- |   |   |
|---|---|
| a) <a href="#">cargo</a> ,                  | a) <a href="#">cargo</a>                  |
| b) <a href="#">destination identifier</a> , | b) <a href="#">destination identifier</a> |
| c) <a href="#">end-of-packet marker</a> ,   | c) <a href="#">end-of-packet marker</a>   |
| d) (SpaceWire) <a href="#">link</a> ,       | d) (SpaceWire) <a href="#">link</a>       |
| e) <a href="#">logical address</a> ,        | e) <a href="#">logical address</a>        |
| f) <a href="#">logical addressing</a> ,     | f) <a href="#">logical addressing</a>     |
| g) <a href="#">path addressing</a> ,        | g) <a href="#">path addressing</a>        |
| h) <a href="#">regional addressing</a> ,    | h) <a href="#">regional addressing</a>    |
| i) (SpaceWire) <a href="#">network</a> ,    | i) (SpaceWire) <a href="#">network</a>    |
| j) (SpaceWire) <a href="#">node</a> , and   | j) (SpaceWire) <a href="#">node</a>       |
| k) (SpaceWire) <a href="#">packet</a> .     | k) (SpaceWire) <a href="#">packet</a>     |

## 1.6.2. TERMS DEFINED IN THE OSI BASIC REFERENCE MODEL

### OSI BASIC REFERENCE MODEL が定義する用語

This document adopts the following terms defined in “Open Systems Interconnection – Basic Reference Model: The Basic Model” [A2]:

- a) (N)-entity (for brevity, this is called “entity” in this document),
- b) flow control,
- c) multiplexing,
- d) protocol data unit,
- e) reassemble,
- f) segmenting,
- g) service, and
- h) service data unit.

本書は、“Open Systems Interconnection – Basic Reference Model: The Basic Model” [A2] が定義する次の用語を採用する。

- a) (N)-構成要素（簡潔にするために、本書では「構成要素」と呼ぶ）
- b) flow control（フロー制御）
- c) multiplexing（多重化）
- d) protocol data unit（プロトコルデータ単位）
- e) reassemble（組立て）
- f) segmenting（分割）
- g) service（サービス）
- h) service data unit（サービスデータ単位）

### 1.6.3. TERMS DEFINED IN THE OSI SERVICE DEFINITION CONVENTIONS OSI SERVICE DEFINITION CONVENTIONS が定義する用語

This document adopts the following terms defined in “Information Technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services” [A3]:

本書は、“Information Technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services” [A3] が定義する次の用語を採用する。

- a) indication,
- b) primitive,
- c) procedure, and
- d) request.

- a) indication
- b) primitive
- c) procedure
- d) request

#### 1.6.4. TERMS DEFINED IN THIS DOCUMENT // 本書が定義する用語

The following definitions are used throughout this document. 本書は次の定義を用いる。

**Packet (SpW-R Packet):** A protocol data unit of SpW-R.

**Packet (SpW-R Packet):** SpW-R の protocol data unit。

**receive:** An action to receive a service data unit or protocol data unit.

**受信する:** service data unit または protocol data unit を受信する動作。

**retransmit:** An action to retransmit a protocol data unit.

**再送信する:** protocol data unit を再送信する動作。

**Receive TEP:** The receiving end of a Transport Channel (i.e. the TEP which receives forward direction protocol data units). A node has zero or more Receive TEPs.

**受信 TEP:** Transport Channel の受信端（つまり、順方向 protocol data units を受信する TEP）。Node は、ゼロ個以上の受信 TEPs を持つ。

**send:** An operation to send a service data unit.

**送る:** service data unit を送る動作。

**transmit:** An operation to transmit a protocol data unit (n.b., “transmit” does not include “retransmit”).

**送信する:** protocol data unit を送信する動作（注：「送信する」は「再送信する」を含まない）。

**Transmit TEP:** The sending end of a Transport Channel (i.e. the TEP which transmits forward direction protocol data units). A node has zero or more Transmit TEPs.

**送信 TEP:** Transport Channel の送信端（つまり、順方向 protocol data units を送信する TEP）。Node は、ゼロ個以上の送信 TEPs を持つ。

**Transport Channel:** A one-way logical data path between two nodes. Zero or more Transport Channels exist between any pair of nodes but only zero or one Transport Channel exists between any pair of a Transmit TEP and a Receive TEP at a certain timing.

**Transport Channel:** 二つの nodes 間の一方方向論理データパス。Nodes の任意のペアの間にはゼロ個以上の Transport Channels が存在するが、送信 TEP と受信 TEP の任意のペアの間には、ある時点において、ゼロ個か一つのみの Transport Channel が存在する。

**Transport End Point (TEP):** A point in a node that sends or receives application data through a Transport Channel over a SpaceWire network. A TEP is either a Transmit TEP or a Receive TEP.

**Transport End Point (TEP):** SpaceWire network 上の Transport Channel を介してアプリケーションデータを送るまたは受信する node 内のポイント。TEP は、送信 TEP か受信 TEP の何れかである。

### 1.6.5. NOTATIONS // 表記

The following notations are used throughout this document. 本書は次の表記を用いる。

A paragraph that begins with “[Example]” (or “[Example  $n$ ]”, where  $n$  is a positive integer) presents an example that is aimed to help readers to understand the specification and is not a part of the specification. “[例]”（または “[例  $n$ ]”、 $n$  は正の整数）で始まる段落は、読者の仕様の理解を助けるための例であり、仕様の一部ではない。

A paragraph that begins with “[Note]” (or “[Note  $n$ ]”, where  $n$  is a positive integer) is an informative note that is aimed to help readers to understand the specification and is not a part of the specification. “[注]”（または “[注  $n$ ]”、 $n$  は正の整数）で始まる段落は、読者の仕様の理解を助けるための付加情報を記したものであり、仕様の一部ではない。

### 1.6.6. CONVENTIONS // 規則

In accordance with the [SpaceWire](#) protocol, data fields are grouped into a series of eight-bit ‘words’. Throughout this document, the unit of an eight-bit word is referred to as an ‘[octet](#)’.

In this document, the following conventions are used to identify each [octet](#) in an [N-octet](#) field. The first [octet](#) in the field to be transmitted (*i.e.*, the leftmost part in associated diagrams, if given) is defined as ‘[octet 0](#)’, the next [octet](#) is defined as ‘[octet 1](#)’, and so on up to ‘[octet N-1](#)’. When a field is used to express a binary value (such as a counter), the Most Significant ([MS](#)) [octet](#) **shall** be the first transmitted [octet](#) of the field, *i.e.*, ‘[octet 0](#)’ (see Figure 1-1).

[SpaceWire](#) プロトコルに則り、データフィールドは、8ビットワードの連なりを一区切りとして扱う。本書では、この8ビットワードの単位を‘[octet](#)’と称する。

本書では、[N-octet](#) のフィールドの各 [octet](#) を特定するため、次の規則を用いる。フィールドの中で伝送する最初の [octet](#) (つまり、図示する場合、図中の最も左側) を‘[octet 0](#)’と定義する。以下、次の [octet](#) を‘[octet 1](#)’というように‘[octet N-1](#)’まで定義する。フィールドがバイナリ値 (カウンタ等) を表わす場合、最上位 ([MS](#)) [octet](#) はフィールドの最初に伝送する [octet](#), つまり、‘[octet 0](#)’ (Figure 1-1 参照)である **こと**。

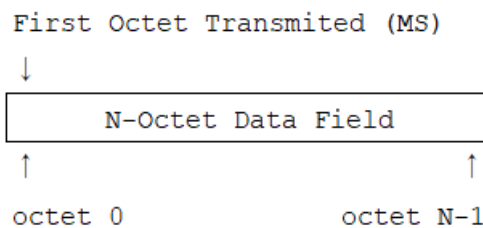
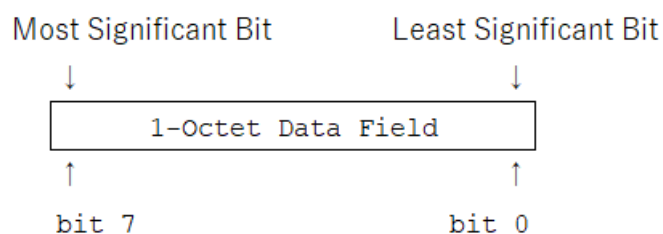


Figure 1-1: [Octet](#) Numbering Convention  
[Octet](#) 番号付け規則



In this document, the Most Significant Bit (MSB) is shown at the leftmost part in an associated diagram, if given. In accordance with the SpaceWire protocol, the following conventions are used to identify each bit in an Octet field. The MSB is defined as 'bit 7', the next bit is defined as 'bit 6', and so on down to 'bit 0'.

本書では、最上位ビット (MSB) を、図示する場合は図中の最も左側に示す。また、SpaceWire プロトコルに則り、Octet のフィールドの各ビットを特定するため、次の規則を用いる。MSB を'bit 7' と定義する。以下、次のビットを'bit 6'というように'bit 0'まで定義する。



**Figure 1-2: Bit Numbering Convention**

### ビット番号付け規則

[Note] The Least Significant bit is the first bit transmitted in the SpaceWire protocol, which is opposite to the convention of the Space Packet protocol [R3].

[注] SpaceWire プロトコルでは、最下位ビットが、初めに送信されるビットである。これは、Space Packet プロトコル [R3] の規則とは逆順である。

## 1.7. VERBAL FORMS // 表現形式

The following conventions apply throughout this document.

- a) the auxiliary verb ‘**shall**’ implies mandatory conditions.
- b) the auxiliary verb ‘**should**’ implies optional but desirable conditions.
- c) the auxiliary verbs ‘**may**’ implies optional conditions.
- d) the auxiliary verb ‘can’ implies capability or ability to do something.
- e) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

The words ‘**shall**’, ‘**should**’, ‘**may**’ are highlighted in **red** and **bold** font.

本書では以下の決まりに従い記述する。

- a) 「...**こと**」「...**なければならない**」は、必須な仕様を示す。
- b) 「...**べき**...」は、任意であるが推奨される仕様を示す。
- c) 「...**良い**...」は、許容される仕様を示す。
- d) 「...できる...」は、何かをする事が可能な事を示す。
- e) 他のパターンの記述は、事実を示す文である。

「...**こと**」「...**なければならない**」「...**べき**...」「...**良い**...」は読者の仕様の理解の助けのため、**赤字・太字**で示す。

[注] 本書では、要求事項を電子的に検索しやすいように、英文の ‘**shall**’ の訳語として、「**こと**」を使用している。逆に、‘**shall**’ の訳語以外では「こと」は使用せず、「事」を用いている。また、英文の ‘**may**’ に対応する訳語として、「**良い**」という当て字を使用している。逆に、‘**may**’ の訳語以外で「良い」は使用していない。

「A, B, 及び C」という表記は、英文の ‘A, B, and C’ に対応し、「A 及び B 及び C」である事を意味する。

「A, B, または C」という表記は、英文の ‘A, B, or C’ に対応し、「A または B または C」である事を意味する。

When a translation into Japanese is provided, the original English version and its Japanese translation are given in the left and right sides, respectively, in principle, as in this paragraph. In some cases, *e.g.*, titles of sections and captions of figures/tables, the English and Japanese versions are put in a single line separated by “//” in this order (“English // Japanese”) or in separate lines with no delimiter in between (“English [Line-Break] Japanese”).

In most cases, the technical terms are not translated into Japanese. The English words in alphabet remain as they are in their Japanese translation. The forms in alphabet in English, which distinguish the singular and plural forms, remain as they are in the Japanese translation to preserve the information of the quantity, although the Japanese language does not inherently distinguish the singular and plural forms.

Technical terms are basically highlighted in **green** and in some cases (names of states and those defined in the other documents) in **blue**. The latter consists of names of documents, protocols, widely used technical terms, and those locally used in some sections (*e.g.*, field names). Note that the first letter of an English word in a technical term is written in the capital letter excluding that in a widely used technical term.

A technical term is underlined as “Term” in each of their first occurrences in chapters 1, 2, and 3 onwards.

日本語への翻訳が存在する場合、原則として、この段落のように、英語を左側に示し、日本語を右側に示す。また、章や図表のタイトル等は、英語、日本語の順に一行中に // で区切る(「英語 // 日本語」)か、二行に分けて区切り文字なし(「英語 [改行] 日本語」)で、記述する場合もある。

多くの場合、技術用語の翻訳は行わず、英単語を維持する。そこで、日本語にもアルファベットが登場する。それらは正本である日本語文中においてもアルファベット表記される。日本語の名詞に単数形、複数形の区別はないが、単複の情報を保つため、日本語文中においても、英語の単数形、複数形の違いはアルファベットでそのまま表記する。

技術用語は読者の便のため基本的に**緑字**、場合(状態の名前と他の文書で定義された用語)により**青字**で示す。後者は、文書名、プロトコル名、広く用いられている技術用語、及び、局所的にしか登場しないもの(フィールド名等)からなる。ここで、技術用語は、広く用いられているものを除き、基本的に大文字始まりの英単語で表記する。

技術用語は、1 章、2 章、3 章以降の各々の初出箇所を「用語」のように下線で明示する。

## 2. OVERVIEW // 概要

This section provides an informative overview of **SpaceWire-R**, which will help the readers to understand the specification given in this document. 本章では、**SpaceWire-R** を概説する。仕様ではないが、本書に記載される仕様の読者の理解に役立つ。

## 2.1. OVERALL PROTOCOL ARCHITECTURE

### 全体的なプロトコルアーキテクチャ

SpaceWire-R is a protocol that provides onboard applications with reliable data transfer services over SpaceWire [A1] networks.

The overall protocol architecture for SpaceWire networks in which SpaceWire-R is used is shown in Figure 2-1. The SpaceWire protocol specified in [A1] is used at the bottom of the protocol stack. SpaceWire-R is located in the layer above the SpaceWire protocol and is used by applications in the upper layer.

SpaceWire-R は、SpaceWire [A1] networks 上での信頼性の高いデータ伝送 services を、搭載アプリケーションに提供するプロトコルである。

SpaceWire-R を用いる場合の SpaceWire networks のプロトコルアーキテクチャの全体を Figure 2-1 に示す。[A1] が定める SpaceWire プロトコルは、プロトコルスタックの最下部で用いる。SpaceWire-R は、SpaceWire プロトコルの層の上部に位置し、上位層のアプリケーションが用いる。

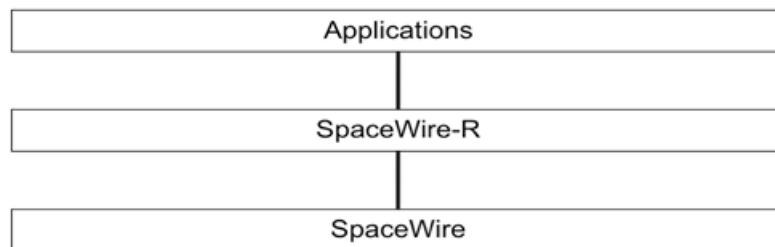


Figure 2-1: Overall Protocol Architecture  
プロトコルアーキテクチャの全体

## 2.2. PROTOCOL FEATURES // プロトコル機能

### 2.2.1. GENERAL // 一般

The primary function of SpaceWire-R (SpW-R) is to transfer data reliably from a sending node to a receiving node over a SpaceWire network. This protocol specifies the following functions:

- a) Multiplexing,
- b) Segmentation (Segmenting),
- c) Data transfer with High Reliability,
- d) Flow Control (optional), and
- e) Heartbeat (optional).

Each of these functions is briefly explained in the following subsections.

SpaceWire-R (SpW-R) の主たる機能は、SpaceWire network 上で、送信 node から 受信 node へ確実にデータを伝送する事である。本プロトコルは以下の機能を定める。

- a) 多重化
- b) Segmentation (Segmenting)
- c) 信頼性の高いデータ伝送
- d) Flow Control (オプション)
- e) Heartbeat (オプション)

これらの機能は、各々、以下の項で簡潔に説明する。

### 2.2.2. MULTIPLEXING // 多重化

SpW-R provides a mechanism for **multiplexing** mutually independent flows of data from a sending **node** identified by a **SpaceWire Logical Address (SLA)** to a receiving **node** identified by another **SLA**. Each of these flows of data is referred to as a **Transport Channel**. Each **Transport Channel** is controlled independently and its parameters to control data flows might vary from a **Transport Channel** to another.

SpW-R は、ある **SpaceWire Logical Address (SLA)** によって特定される送信 **node** から、別の **SLA** によって特定される受信 **node** への相互に独立したデータフローを**多重化**する機構を提供する。これらのデータフローの各々を、**Transport Channel** と称する。各 **Transport Channel** は独立して制御され、データフローを制御するためのパラメータは **Transport Channels** ごとに異なる事がある。

### 2.2.3. SEGMENTATION (SEGMENTING)

If the size of a data unit sent by the sending application (a **service data unit**) does not conform to a **protocol data unit** of SpW-R (referred to as a **SpW-R Packet**) that can be transmitted with the **SpaceWire** protocol, the **service data unit** is split into smaller pieces (*i.e.* **segmenting** is performed) at the sending end (referred to as the **Transmit TEP**) so that each piece can be transmitted in a **SpW-R Packet**. At the receiving end (referred to as the **Receive TEP**), the original **service data unit** is reconstructed from a series of the received **SpW-R Packets** (*i.e.* **reassemble** is performed) and is delivered to the receiving application.

送信アプリケーションが送るデータ単位 (**service data unit**) の大きさが、**SpaceWire** プロトコルで送信できる SpW-R の **protocol data unit** (**SpW-R Packet** と称する) に適合しない場合、送信端 (**送信 TEP** と称する) では、**segmenting** を実施する。つまり、**service data unit** をより小さな断片に分割する事で、各断片が一つの **SpW-R Packet** で送信できるようにする。受信端 (**受信 TEP** と称する) では、**reassemble** を実施、つまり、一連の受信 **SpW-R Packets** から、元の **service data unit** を再構築し、受信アプリケーションに配信する。

## 2.2.4. DATA TRANSFER WITH HIGH RELIABILITY // 信頼性の高いデータ伝送

The function of “Data Transfer with High Reliability” guarantees transfer of data from the sending application to the receiving application without loss or duplication while keeping the sequence identical to that in which the sending application has sent. For this purpose, in this protocol, data are retransmitted when the report of the reception of the data is not obtained from the receiver.

In SpW-R, a sliding window is used to control a retransmission of Data Packets. The range of the sliding window of SpW-R is slid so that the oldest Data Packet (or the Data Ack Packet for the Data Packet) among those whose Sequence Numbers are contained in the sliding window is marked as not yet received.

[Note] In convention, the sliding window refers to the range of the Sequence Numbers of Packets for which the information on whether the corresponding Ack Packets have been received or not is held. The corresponding Ack Packets are guaranteed to have been received for all Packets whose Sequence Numbers are not included within the sliding window.

The SpW-R protocol itself does not have an ability to respond the occurrence of a failure in a Transport Channel. If redundancy control is needed for a Transport Channel, the redundancy control must be executed by the application. This document also describes what is needed for the redundancy control.

「信頼性の高いデータ伝送」の機能は、送信アプリケーションから受信アプリケーションへ伝送したデータにつき、損失なく、重複せずに、送信アプリケーションが送信したのと同じの順序を保持する事を保証する。その目的のために、本プロトコルでは、受信者からデータの受信の報告が得られない場合、そのデータは再送信される。

SpW-R では、Data Packets の再送制御に sliding window を用いる。SpW-R の sliding window の範囲は、sliding window に含まれる Sequence Numbers の中で最古の Data Packet (あるいはその Data Packet に対する Data Ack Packet) のものが未受信と印されるようにスライドされる。

[注] 一般的に、sliding window は、対応する Ack Packets が受信されたか否かの情報を保持する Packets の Sequence Numbers の範囲を指し示す。Sequence Numbers が sliding window に含まれていない全ての Packets に対して、対応する Ack Packets が受信されている事が保証される。

SpW-R プロトコルは、それ自身、Transport Channel に障害の発生に対応する機能を持たない。そこで、ある Transport Channel に対して冗長系制御が必要な場合、冗長系制御はアプリケーションが実行する必要がある。本書には冗長系制御に必要な事項も記す。



### 2.2.5. FLOW CONTROL

SpW-R provides a mechanism of flow control, referred to as Flow Control, with which the sending entity of the protocol does not send data that would exceed the receiving capability of the receiving entity. In Flow Control, the receiving entity tells the sending entity how many more data units it can receive. The use of Flow Control is optional for each Transport Channel.

SpW-R は、プロトコルの送信構成要素が受信構成要素の受信能力を超えるデータを送らないようにする事を実現する、Flow Control と称する flow control の機構を提供する。Flow Control では、受信構成要素が、送信構成要素に、あと幾つのデータ単位を受信できるかを通知する。Flow Control の使用は各 Transport Channel についてオプションである。

### 2.2.6. HEARTBEAT

SpW-R provides a mechanism for the sending and receiving entities of the protocol to check whether the link and the other entity are still alive even when there are no data to send or receive, respectively. The mechanism is referred to as Heartbeat. The use of Heartbeat is optional for each Transport Channel.

SpW-R は、プロトコルの送信及び受信構成要素に、それぞれ、送信または受信するデータがないときも含めて、link と他方の構成要素がまだ生きているかどうかを確認するための、Heartbeat と称する機構を提供する。Heartbeat の使用は、各 Transport Channel についてオプションである。

## 2.3. RELATIONSHIP WITH OTHER PROTOCOLS // 他のプロトコルとの関係

This protocol is based on the protocol called the [Joint Architecture Standard \(JAS\) Reliable Data Delivery Protocol \(RDDP\)](#) [R1] developed by the [Sandia National Laboratories](#). The [JAS RDDP](#) is based on the protocol called the [GOES-R Reliable Data Delivery Protocol \(GRDDP\)](#) [R2] developed by the [Goddard Space Flight Center \(GSFC\)](#) of [NASA](#). The differences between the [JAS RDDP](#) and the [GRDDP](#) are described in [R1].

The technical differences between this protocol and the [JAS RDDP](#) are as follows:

本プロトコルは、[Sandia National Laboratories](#) によって開発された [Joint Architecture Standard \(JAS\) Reliable Data Delivery Protocol \(RDDP\)](#) [R1] と呼ばれるプロトコルに基づく。[JAS RDDP](#) は、[NASA](#) の [Goddard Space Flight Center \(GSFC\)](#) によって開発された [GOES-R Reliable Data Delivery Protocol \(GRDDP\)](#) [R2] と呼ばれるプロトコルに基づく。[JAS RDDP](#) と [GRDDP](#) の違いは [R1] に記される。

本プロトコルと [JAS RDDP](#) の技術的な違いは以下のとおりである。

- 1) The capability of sending urgent messages is removed because the function can be fulfilled with another method.
  - 2) The constraint requiring that the size of the [sliding window](#) **shall** be a power of two is removed in order to enhance the flexibility of the protocol.
  - 3) The [Open/Reset Command](#) is renamed the [Open Command](#) in order to simplify the terminology.
  - 4) [Ack Packets](#) for [Control Packets](#) have the different values of [Packet Types](#) from those of the [Ack Packets](#) for [Data Packets](#) in order to differentiate the former from the latter even when both of them have [Sequence Number](#) 0.
  - 5) The way in which [Open Commands](#), [Close Commands](#), and the [Ack Packets](#) for these [Commands](#) are handled is specified in detail in order to prevent potential disagreement between the states of whether the [Transport Channel](#) is open or close as recognized by the [Transmit TEP](#) and [Receive TEP](#). The major changes are:
    - 5a) A [Close Command](#) can be transmitted only when the [Ack Packet](#) for an [Open Command](#) has been received; this constraint is set because a [Close Command](#) is used solely to close an opened [Transport Channel](#).
- 1) 緊急メッセージを送る能力は、その機能を他の方法で達成できるため削除した。
  - 2) プロトコルの柔軟性を高めるために、[sliding window](#) のサイズが 2 の累乗で**なければなら****ない**という制約は取り除いた。
  - 3) 用語を簡略化するために、[Open/Reset Command](#) は、[Open Command](#) に改名した。
  - 4) [Control Packets](#) に対する [Ack Packets](#) と、[Data Packets](#) に対する [Ack Packets](#) とは、前者と後者でそのいずれもが [Sequence Number](#) が 0 の場合でも区別できるように、異なる [Packet Types](#) の値を持つ。
  - 5) [Transport Channel](#) が開いているか閉じているかどうか、[送信 TEP](#) と [受信 TEP](#) とが認識する状態の潜在的な不一致を防ぐために、[Open Commands](#) と [Close Commands](#)、並びに、これらの [Commands](#) に対する [Ack Packets](#) の扱い方を詳細に定めた。主要な変更点は:
    - 5a) [Close Command](#) はオープンした [Transport Channel](#) を閉じるためのみに使われるものである事を考慮して、[Close Command](#) は [Open Command](#) に対する [Ack Packet](#) を受信した場合にのみ送信できるという制約を設けた。

5b) If the Receive TEP has found that the Transport Channel is inactive, it can close the Transport Channel.

6) The mechanism of Flow Control, along with two new types of Packets for it, “Flow Control Packets” and “Flow Control Ack Packets”, are introduced as options for the purpose of enhancing the functionality of the protocol.

7) The mechanism of Heartbeat, along with two new types of Packets for it, “Heartbeat Packets” and “Heartbeat Ack Packets”, is introduced as options for the purpose of enhancing the functionality of the protocol.

If none of the above-described features, which are unique to SpaceWire-R, is used, SpaceWire-R is compatible with JAS RDDP except for one point: the value of the Packet Type fields for Ack Packets for Control Packets must be amended (see Table 4-2).

The style of the document has also been changed from those of [R1] and [R2], and it is now compatible with the style in which the ISO and CCSDS protocol specifications are written except that the Japanese translation is also written and that terms are highlighted with color.

5b) 受信 TEP は、Transport Channel が非アクティブである事を検出すると、Transport Channel を閉じる事ができる。

6) プロトコルの機能を強化するためのオプションとして、Flow Control の機構が、そのための新たな二種類の Packets, “Flow Control Packets” と “Flow Control Ack Packets”, と共に導入された。

7) プロトコルの機能を強化するためのオプションとして、Heartbeat の機構が、そのための新たな二種類の Packets, “Heartbeat Packets” と “Heartbeat Ack Packets”, と共に導入された。

上記の SpaceWire-R に固有の機能を用いない場合、SpaceWire-R は JAS RDDP と互換性がある。ただし、Control Packets に対する Ack Packets の Packet Type フィールドの値を修正する必要がある (Table 4-2 参照)。

文書のスタイルも [R1] 及び [R2] のスタイルから変更された。日本語訳を併記した事と用語に色付けをした事を除き、ISO 及び CCSDS のプロトコル仕様が記述されるスタイルと互換性がある。

### 3. SERVICE DEFINITION // SERVICE の定義

#### 3.1. GENERAL // 一般

This section presents the definition of the SpW-R services in the form of primitives, where the primitive represents an abstract model of logical exchanges of data and control information between the protocol entity of SpaceWire-R and applications that use SpaceWire-R.

The parameters of the primitive specify the information available to applications. This document specifies the meanings of the parameters in an abstract notation. The way in which a specific implementation makes the information available is not constrained by this specification. In addition to the parameters specified in this section, an implementation may provide other parameters to applications (e.g., parameters for controlling services, monitoring performance, and making diagnosis).

本章は、SpW-R の service 定義を、primitives の形式で示す。Primitive は、SpaceWire-R のプロトコル構成要素と SpaceWire-R を用いるアプリケーションとの間の、データと制御情報の論理的なやり取りの抽象モデルを示す。

Primitive の parameters は、アプリケーションで使用可能な情報を指定する。本書は、これらの parameters の意味を抽象化された表記で定める。本仕様は、特定の実装がこの情報を使用可能にする方法を拘束しない。本章が定める parameters に加えて、実装は、アプリケーションに他のパラメータを提供しても良い（例えば、service の制御、パフォーマンスの監視、及び診断するためのパラメータ）。

#### 3.2. CHANNEL CONTROL SERVICE

##### 3.2.1. OVERVIEW OF CHANNEL CONTROL SERVICE CHANNEL CONTROL SERVICE の概要

The Channel Control Service provides each of applications with the capability of controlling the state machines of the Transmit and Receive TEPs of Transport Channels (see Sections 4.4.3 and 4.5.2).

Channel Control Service は、Transport Channels の送信及び受信 TEPs のステートマシンを制御する能力を、各アプリケーションに提供する (4.4.3 項及び 4.5.2 項参照)。

### 3.2.2. CHANNEL CONTROL SERVICE PARAMETERS

#### 3.2.2.1. Channel Number

The parameter [Channel Number](#) (see Section 4.2.3.6) describes the [Transport Channel](#) whose state machine is controlled by an application.

Parameter [Channel Number](#) (4.2.3.6 項参照) は、アプリケーションが制御する対象のステートマシンを持つ [Transport Channel](#) を記す。

#### 3.2.2.2. Directive Type

The parameter [Directive Type](#) describes the type of the directive that the application delivers to the [Transmit or Receive TEP](#) through the [ChannelControl.request primitive](#). The possible values of this parameter are specified in Table 3-1.

Parameter [Directive Type](#) は、アプリケーションが [ChannelControl.request primitive](#) を介して [送信または受信 TEP](#) に配信する指令の種別を記す。この parameter がとり得る値は、Table 3-1 に定める。

Table 3-1: Values for the [Directive Type](#)  
[Directive Type](#) の値

Directive Type	Description	Note
Open (Transport Channel)	Directs a <a href="#">Transmit or Receive TEP</a> to open the specified <a href="#">Transport Channel</a> . <a href="#">送信または受信 TEP</a> に指定した <a href="#">Transport Channel</a> のオープンを指示する	Used by the applications at the sending and receiving ends of the <a href="#">Transport Channel</a> . <a href="#">Transport Channel</a> の送信及び受信端のアプリケーションが用いる。
Close (Transport Channel)	Directs a <a href="#">Transmit TEP</a> to close the specified <a href="#">Transport Channel</a> . <a href="#">送信 TEP</a> に指定した <a href="#">Transport Channel</a> のクローズを指示する	Used only by the application at the sending end of the <a href="#">Transport Channel</a> . <a href="#">Transport Channel</a> の送信端のアプリケーションのみが用いる。

### 3.2.2.3. Notification Type

The parameter [Notification Type](#) describes the event of which the [Transmit or Receive TEP](#) notifies the application through the [ChannelControl.indication primitive](#). The possible values of this [parameter](#) are specified in Table 3-2.

Parameter [Notification Type](#) は、送信または受信 TEP が [ChannelControl.indication primitive](#) を介してアプリケーションに通知するイベントを記す。この [parameter](#) がとり得る値は、Table 3-2 に定める。

Table 3-2: Values for the [Notification Type](#)  
[Notification Type](#) の値

Notification Type	Description
Enabled	The <a href="#">Transmit or Receive TEP</a> has transitioned into the <a href="#">ENABLED</a> state. <a href="#">Transmit または Receive TEP</a> が、 <a href="#">ENABLED</a> 状態に移行した。
Open	The <a href="#">Transmit or Receive TEP</a> has transitioned into the <a href="#">OPEN</a> state. <a href="#">Transmit または Receive TEP</a> が、 <a href="#">OPEN</a> 状態に移行した。
Closing	The <a href="#">Transmit or Receive TEP</a> has transitioned into the <a href="#">CLOSING</a> state. <a href="#">Transmit または Receive TEP</a> が、 <a href="#">CLOSING</a> 状態に移行した。
Closed	The <a href="#">Transmit or Receive TEP</a> has transitioned into the <a href="#">CLOSED</a> state. <a href="#">Transmit または Receive TEP</a> が、 <a href="#">CLOSED</a> 状態に移行した。

### 3.2.3. CHANNEL CONTROL SERVICE PRIMITIVES

#### 3.2.3.1. General // 一般

The **service primitives** associated with this **service** are as follows:

- 1) ChannelControl.request, and
- 2) ChannelControl.indication.

The ChannelControl.request **primitive** passes information from an application to a **Transmit or Receive TEP** in order to request that the state of the **Transmit or Receive TEP**, respectively, be transitioned.

The ChannelControl.indication **primitive** passes information from the **Transmit or Receive TEP** to the application in order to notify that the state of the **Transmit or Receive TEP**, respectively, has transitioned.

本 **service** に関する **service primitives** は、以下のとおりである。

- 1) ChannelControl.request
- 2) ChannelControl.indication

ChannelControl.request **primitive** は、送信または受信 TEP の状態の遷移を要求するために、アプリケーションから、それぞれ、送信または受信 TEP に情報を引き渡す。

ChannelControl.indication **primitive** は、送信または受信 TEP の状態が遷移した事を通知するために、それぞれ、送信または受信 TEP からアプリケーションに情報を引き渡す。

#### 3.2.3.2. ChannelControl.request

The ChannelControl.request **primitive shall** be used by an application for requesting the **Transmit or Receive TEP** of the specified **Transport Channel** to change its state (see Sections 4.4.3 and 4.5.2).

The ChannelControl.request **primitive shall** take the following **parameters** in the following form:

ChannelControl.request( **Channel Number**,  
**Directive Type**)

ChannelControl.request **primitive** は、アプリケーションが、指定する **Transport Channel** の送信または受信 TEP に対してその状態の変更を、その TEP に要求するために用いること (4.4.3 項及び 4.5.2 項参照)。

ChannelControl.request **primitive** は、以下の **parameters** を以下のように受け付けること。

#### 3.2.3.3. ChannelControl.indication

The ChannelControl.indication **primitive shall** be used by a **Transmit or Receive TEP** for notifying the application that a state transition has occurred.

The ChannelControl.indication **primitive shall** take the following **parameters** in the following form:

ChannelControl.indication( **Channel Number**,  
**Notification Type**)

ChannelControl.indication **primitive** は、送信または受信 TEP が、状態遷移の発生を、アプリケーションに通知するために用いること。

ChannelControl.indication **primitive** は、以下の **parameters** を以下のように受け付けること。

### 3.3. DATA TRANSFER SERVICE // データ伝送 SERVICE

#### 3.3.1. OVERVIEW OF DATA TRANSFER SERVICE // データ伝送 SERVICE の概要

The Data Transfer Service provides each of applications with transfer of sequences of service data units. One sequence of service data units of variable length is transferred through one Transport Channel over a SpaceWire network. The transfer of service data units in a Transport Channel is unidirectional (one way) and asynchronous.

データ伝送 Service は、service data units の並びの伝送を、各アプリケーションに提供する。SpaceWire network 上の一つの Transport Channel を介して、可変長の service data units の並びが一つ伝送される。ある Transport Channel での Service data units の伝送は、単方向（一方向）かつ非同期である。



### 3.3.2. DATA TRANSFER SERVICE PARAMETERS データ伝送 SERVICE PARAMETERS

#### 3.3.2.1. Service Data Unit

The [parameter Service Data Unit \(SDU\)](#) describes a variable-length, [octet-aligned service data unit](#), the format of which is unknown to the [protocol entity](#). A [Service Data Unit](#) is provided by a sending application to a [Transmit TEP](#) and is delivered to the receiving application by the [Receive TEP](#). The [service data unit](#) is carried from the [Transmit TEP](#) to the [Receive TEP](#) in the [Application Data](#) field (see Section 4.2.4.3) of a [Data Packet](#) (or a series of [Data Packets](#)).

The maximum length of a [service data unit](#) **shall** be determined for each [Transport Channel](#) and **shall** be specified in a project-specific document using the table provided in Section 5.2 of this document.

#### 3.3.2.2. SDU ID

The [parameter Service Data Unit Identifier \(SDU ID\)](#) describes the sequence number of the [service data unit](#) that the sending application provides through the [DataTransfer.request primitive](#). This number is used in the subsequent [DataTransferNotify.indication primitive](#) to identify the [service data unit](#) associated with the [primitive](#).

#### 3.3.2.3. Channel Number

The [parameter Channel Number](#) (see Section 4.2.3.6) specifies the [Transport Channel](#) in which the [service data unit](#) is transferred.

[Parameter Service Data Unit \(SDU\)](#) は、可変長の [octet](#) 単位の [service data unit](#) を記し、[プロトコル構成要素](#)はその形式を関知しない。[Service Data Unit](#) は、送信アプリケーションが[送信 TEP](#) に提供し、[受信 TEP](#) が受信アプリケーションに配信する。[Service data unit](#) は、[Data Packet](#) (または一連の [Data Packets](#)) の [Application Data](#) フィールド (4.2.4.3 項参照) にて、[送信 TEP](#) から[受信 TEP](#) に運ばれる。

[Service data unit](#) の最大長は、[Transport Channel](#) 毎に決めること。また、本書 5.2 項に掲載する表を用いてプロジェクト固有の文書に定めること。

[Parameter Service Data Unit Identifier \(SDU ID\)](#) は、送信アプリケーションが [DataTransfer.request primitive](#) を介して提供する [service data unit](#) のシーケンス番号を記す。この番号は、引き続く [DataTransferNotify.indication primitive](#) で、その [primitive](#) に関する [service data unit](#) を特定するために用いられる。

[Parameter Channel Number](#) (4.2.3.6 項参照) は、[service data unit](#) を伝送する [Transport Channel](#) を指定する。

### 3.3.2.4. Notification Type

The parameter [Notification Type](#) describes the event of which the [Transmit TEP](#) notifies the sending application through the `DataTransferNotify.indication` [primitive](#). The possible values of this parameter are specified in Table 3-3.

Parameter [Notification Type](#) は、送信 TEP が `DataTransferNotify.indication` [primitive](#) を介して送信アプリケーションに通知するイベントを記す。この parameter がとり得る値は、Table 3-3 に定める。

Table 3-3: Values for the [Notification Type](#)  
[Notification Type](#) の値

Notification Type	Description
Accept Transfer	The <a href="#">Transmit TEP</a> has accepted a transfer of the <a href="#">Service Data Unit</a> . 送信 TEP が <a href="#">Service Data Unit</a> の伝送を受理した。
Reject Transfer	The <a href="#">Transmit TEP</a> has rejected a transfer of the <a href="#">Service Data Unit</a> . 送信 TEP が <a href="#">Service Data Unit</a> の伝送を拒否した。
Transfer Confirmed	A transfer of the <a href="#">Service Data Unit</a> has been confirmed by the <a href="#">Receive TEP</a> . <a href="#">Service Data Unit</a> の伝送が、受信 TEP により確認された。
Transfer Failure	A transfer of the <a href="#">Service Data Unit</a> has not been confirmed by the <a href="#">Receive TEP</a> . <a href="#">Service Data Unit</a> の伝送が、受信 TEP により確認されなかった。

### 3.3.2.5. Reason

The parameter [Reason](#) describes the reason for the event of which the [Transmit TEP](#) notifies the sending application through the `DataTransferNotify.indication` primitive when the [Notification Type](#) is “[Reject Transfer](#)” (see Section 3.3.2.4). The possible values of this parameter are specified in Table 3-4. When the [Notification Type](#) is not “[Reject Transfer](#)”, the value of this parameter is not defined.

Parameter [Reason](#) は、[Notification Type](#) が “[Reject Transfer](#)” (3.3.2.4 項参照) の場合に、[送信 TEP](#) が `DataTransferNotify.indication` primitive を介して送信アプリケーションに通知するイベントの理由を記す。この parameter がとり得る値は、Table 3-4 に定める。[Notification Type](#) が “[Reject Transfer](#)” でない場合、この parameter の値は定義しない。

Table 3-4: Values for the [Reason](#)

#### [Reason](#) の値

Reason	Description
SDU too long	The length of the <a href="#">Service Data Unit</a> exceeds the maximum length of a <a href="#">Service Data Unit</a> (see Section 4.4.2). <a href="#">Service Data Unit</a> の長さが、 <a href="#">Service Data Unit</a> の最大長を超えた (4.4.2 項参照)。
Channel Not Open	The <a href="#">Transport Channel</a> is not <a href="#">OPEN</a> (see Section 4.4.3). <a href="#">Transport Channel</a> が <a href="#">OPEN</a> されていない (4.4.3 項参照)。

### 3.3.3. DATA TRANSFER SERVICE PRIMITIVES // データ伝送 SERVICE PRIMITIVES

#### 3.3.3.1. General // 一般

The **service primitives** associated with this **service** are as follows.

- 1)     **DataSetTransfer.request**
- 2)     **DataSetTransferNotify.indication**
- 3)     **DataSetTransfer.indication**

The **DataSetTransfer.request primitive** passes information from the sending application to the **Transmit TEP** in order to request that a **service data unit** be transferred to the receiving application through the specified **Transport Channel**.

The **DataSetTransferNotify.indication primitive** passes information from the **Transmit TEP** to the sending application in order to notify the sending application of the completion of the transfer of a **service data unit**.

The **DataSetTransfer.indication primitive** passes information from the **Receive TEP** to the receiving application in order to deliver a received **service data unit** to the receiving application.

本 **service** に関する **service primitives** は、以下のとおりである。

- 1)     **DataSetTransfer.request**
- 2)     **DataSetTransferNotify.indication**
- 3)     **DataSetTransfer.indication**

**DataSetTransfer.request primitive** は、指定した **Transport Channel** を介して **service data unit** を受信アプリケーションに伝送するように要求するために、送信アプリケーションから**送信 TEP** に情報を引き渡す。

**DataSetTransferNotify.indication primitive** は、ある **servicedata unit** の伝送の完了を送信アプリケーションに通知するために、**送信 TEP** から送信アプリケーションに情報を引き渡す。

**DataSetTransfer.indication primitive** は、受信した **service data unit** を受信アプリケーションに配信するために、**受信 TEP** から受信アプリケーションに情報を引き渡す。

#### 3.3.3.2. DataSetTransfer.request

The **DataSetTransfer.request primitive shall** be used by the sending application for requesting the **Transmit TEP** to send a **service data unit** through the specified **Transport Channel**.

The **DataSetTransfer.request primitive shall** take the following **parameters** in the following form:

**DataSetTransfer.request**(   **Service Data Unit**,  
                              **SDU ID**,  
                              **Channel Number**)

**DataSetTransfer.request primitive** は、指定した **Transport Channel** で **service data unit** を送るように、送信アプリケーションが**送信 TEP** に要求するために用いる**こと**。

**DataSetTransfer.request primitive** は、以下の **parameters** を以下のように受け付ける**こと**。

### 3.3.3.3. DataTransferNotify.indication

The DataTransferNotify.indication **primitive shall** be used by the **Transmit TEP** for notifying the sending application of the event associated with the transfer of the **service data unit** identified by the specified **SDU ID**.

DataTransferNotify.indication **primitive** は、指定した **SDU ID** によって特定される **service data unit** の伝送に関するイベントを、**送信 TEP** が送信アプリケーションに通知するために用いる**こと**。

The DataTransferNotify.indication **primitive shall** take the following **parameters** in the following form:

DataTransferNotify.indication **primitive** は、以下の **parameters** を以下のように受け付ける**こと**。

```
DataTransferNotify.indication( SDU ID,  
                                Channel Number,  
                                Notification Type,  
                                Reason)
```

### 3.3.3.4. DataTransfer.indication

The DataTransfer.indication **primitive shall** be used by the **Receive TEP** for delivering a **service data unit** received in the **Transport Channel** to the receiving application.

DataTransfer.indication **primitive** は、**Transport Channel** で受信した **service data unit** を、**受信 TEP** が受信アプリケーションに配信するために用いる**こと**。

The DataTransfer.indication **primitive shall** take the following **parameters** in the following form:

DataTransfer.indication **primitive** は、以下の **parameters** を以下のように受け付ける**こと**。

```
DataTransfer.indication( Service Data Unit,  
                          Transport Channel)
```

## 4. PROTOCOL SPECIFICATION // プロトコル仕様

### 4.1. GENERAL // 一般

This section specifies **SpaceWire-R (SpW-R)** in terms of the **protocol data units** exchanged between paired **protocol entities** (see Section 4.2) and of the **procedures** conducted by the **protocol entities** (see Sections 4.3, 4.4, 4.5, and 4.6).

本章は、組となる**プロトコル構成要素**の間で交換される **protocol data units** (4.2 項参照) と、それらの**プロトコル構成要素**が実行する **procedures** (4.3 項、4.4 項、4.5 項、及び 4.6 項参照) との観点から **SpaceWire-R (SpW-R)** を定める。

## 4.2. SPW-R PACKETS

The **protocol data unit (PDU)** of **SpaceWire-R** is referred to as the **SpW-R Packet**.

In the **SpaceWire** protocol, an **address** is used to uniquely identify the delivery route and destination **node** of a **SpW packet**. The **address** consists of **destination identifiers** and an **SpaceWire Logical Address (SLA)**. The **SpaceWire** protocol uses a variable-length **address** to accommodate the **networks** that require multiple address **octets**, such as **octets** in **SpaceWire path addressing** or **regional addressing**.

The values of the **destination identifiers** and **SLA** are assigned according to the rules of the **logical addresses** specified in [A1]. In the **SpW-R** protocol, the **address** **shall** be within the constraint of the maximum length of 16 **octets**.

[Note] In the case where only **logical addressing** is used in a **SpaceWire network**, the **destination identifiers** are absent in the **SpaceWire network**.

The **destination identifiers** and **SLA** of the **Receive TEP** and **Transmit TEP** **shall** be determined for each **Transport Channel** and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

**SpaceWire-R** の **protocol data unit (PDU)** は、**SpW-R Packet** と称する。

**SpaceWire** プロトコルでは、**SpW packet** の配信経路と配信先 **node** を一意に特定するために **address** を用いる。この **address** は、**destination identifiers** と **SpaceWire Logical Address (SLA)** からなる。**SpaceWire** プロトコルは、**SpaceWire path addressing** や **regional addressing** による **octets** 等、複数 **octets** の **address** を必要とする **networks** に対応する可変長の **address** を用いる。

**Destination identifiers** と **SLA** の値は、[A1] が定める **logical addresses** の規則に従って割り当てる。**SpW-R** プロトコルでは、**address** は、最大長 16 **octets** の制約の下にあること。

[注] ある **SpaceWire network** で **logical addressing** のみを用いる場合、その **SpaceWire network** では **destination identifiers** は存在しない。

受信 TEP 及び送信 TEP の **destination identifiers** 及び **SLA** は、**Transport Channel** 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

#### 4.2.1. TYPES OF SPW-R PACKETS // SPW-R PACKETS の種類

There are five types of SpW-R Packets:

- a) Data Packets,
- b) Control Packets,
- c) Ack Packets,
- d) Flow Control Packets, and
- e) Heartbeat Packets.

SpW-R Packets には以下の五種類がある。

- a) Data Packets
- b) Control Packets
- c) Ack Packets
- d) Flow Control Packets
- e) Heartbeat Packets

##### 4.2.1.1. Data Packets

Data Packets are used by the Transmit TEP of a Transport Channel to send service data units to the Receive TEP of the Transport Channel.

Data Packets は、ある Transport Channel の送信 TEP が、その Transport Channel の受信 TEP に、service data units を送るために用いる。

##### 4.2.1.2. Control Packets

Control Packets are used by a Transmit TEP to send Control Commands to the Receive TEP to control the state machine of the Receive TEP (see Section 4.5.2).

Control Packets は、送信 TEP が、受信 TEP のステートマシンを制御するために制御コマンドを受信 TEP に送るために用いる (4.5.2 項参照)。

There are two types of Control Commands:

二種類の制御コマンドがある。

- a) Open Command: Open Commands are used to change the state of the Receive TEP from the ENABLED state to the OPEN state (see Section 4.5.2.3), and
- b) Close Command: Close Commands are used to change the state of the Receive TEP from the OPEN state to the CLOSING state (see Section 4.5.2.4).

- a) Open Command: Open Commands は、受信 TEP の状態を、ENABLED 状態から OPEN 状態に変えるために用いる (4.5.2.3 項参照);。
- b) Close Command: Close Commands は、受信 TEP の状態を、OPEN の状態から CLOSING 状態に変えるために用いる (4.5.2.4 項参照)。



### 4.2.1.3. Ack Packets

Ack Packets are used by a Receive TEP to notify the Transmit TEP of receipt of Data, Control, and Heartbeat Packets (and the MASN in some cases; see the last paragraph in this section) and also by a Transmit TEP to notify the Receive TEP of receipt of Flow Control and Heartbeat Packets.

There are four types of Ack Packets:

- a) Data Ack Packet: used to notify receipt of a Data Packet,
- b) Control Ack Packet: used to notify receipt of a Control Packet,
- c) Flow Control Ack Packet: used to notify receipt of a Flow Control Packet, and
- d) Heartbeat Ack Packet: used to notify receipt of a Heartbeat Packet.

In the case where Flow Control is used, Data Ack and Control Ack Packets are also used by a Receive TEP to notify the Transmit TEP of the Maximum Acceptable Sequence Number (MASN) (see Section 4.2.4.4).

### 4.2.1.4. Flow Control Packets

In the case where Flow Control is used, a Flow Control Packet is used by a Receive TEP to notify the Transmit TEP of the MASN (see Section 4.5.3.5) when there is neither a Control Ack Packet nor Data Ack Packet to be transmitted.

[Note] The MASN is also notified to the Transmit TEP by using a Data Ack or Control Ack Packet (see Section 4.2.1.3).

### 4.2.1.5. Heartbeat Packets

Heartbeat Packets are used by the Transmit TEP and/or the Receive TEP of a Transport Channel to solicit the counterpart Receive TEP and/or Transmit TEP, respectively, for a response in order to confirm that the link and the TEP at the other end are alive.

Ack Packets は、受信 TEP が、Data, Control, 及び Heartbeat Packets を受信した事を、(本項の最終段落で述べるようにある条件下では MASN と共に) 送信 TEP に通知するために用いる。また、送信 TEP が Flow Control 及び Heartbeat Packets を受信した事を、受信 TEP に通知するためにも用いる。

Ack Packets には以下の四種類がある。

- a) Data Ack Packet: Data Packet の受信の通知に用いる
- b) Control Ack Packet: Control Packet の受信の通知に用いる
- c) Flow Control Ack Packet: Flow Control Packet の受信の通知に用いる
- d) Heartbeat Ack Packet: Heartbeat Packet の受信の通知に用いる

Flow Control を用いる場合、Data Ack 及び Control Ack Packets は、受信 TEP が、Maximum Acceptable Sequence Number (MASN) (4.2.4.4 項参照) を、送信 TEP に通知するためにも用いる。

Flow Control Packet は、Flow Control を用いる場合において、送信する Control Ack Packet も Data Ack Packet の何れもない時に、受信 TEP が MASN を送信 TEP に通知するために用いる (4.5.3.5 項参照)。

[注] MASN は、Data Ack または Control Ack Packet を用いても、送信 TEP に通知される (4.2.1.3 項参照)。

Heartbeat Packets は、ある Transport Channel の送信 TEP 及び/または受信 TEP が、link ともう一方の端の TEP とが生きているかどうかの確認のため、それぞれ、対応する受信 TEP 及び/または送信 TEP から応答を求めるために用いる。

## 4.2.2. STRUCTURE OF THE SPW-R PACKET // SPW-R PACKET 構造

### 4.2.2.1. SpW packet Structure // SpW packet 構造

A SpW-R Packet shall be the cargo of a SpaceWire packet (see Section 9.2.1 of [A1]).

SpW-R Packet は、SpaceWire packet の cargo ([A1] の 9.2.1 項を参照) であること。

According to Sections 9.2.1 and 9.2.2 of [A1], a SpW-R Packet is preceded by a list of zero or more destination identifiers. The destination identifiers of a forward and reverse direction SpW-R Packet shall be destination identifiers of the Receive TEP and Transmit TEP (see Section 4.2), respectively.

[A1] の 9.2.1 項及び 9.2.2 項に従って、ゼロ個以上の destination identifiers のリストが SpW-R Packet の前にある。順方向及び逆方向の SpW-R Packet の destination identifiers は、それぞれ、受信 TEP 及び送信 TEP の destination identifiers (4.2 項参照) であること。

According to Sections 9.2.1 and 9.2.3 of [A1], a SpW-R Packet is followed by an end-of-packet marker.

[A1] の 9.2.1 項及び 9.2.3 項に従って、end-of-packet marker は SpW-R Packet の後に続く。

The structure of a SpW packet that carries a SpW-R Packet is shown in Figure 4-1.

SpW-R Packet を運ぶ SpW packet の構造を Figure 4-1 に示す。

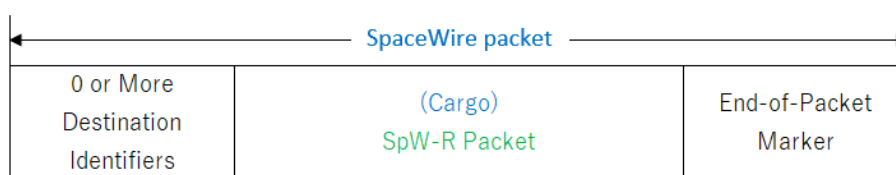


Figure 4-1: Structure of a SpW packet containing a SpW-R Packet  
SpW-R Packet を含む SpW packet の構造

#### 4.2.2.2. SpW-R Packet Structure // SpW-R Packet 構造

A SpW-R Packet shall consist of the following fields, placed contiguously in the following order:

SpW-R Packet は、以下の順序で連続して配置された、以下のフィールドからなること。

- a) Header (10+N octets),
- b) Payload (X octets), and
- c) Trailer (2 octets).

- a) Header (10 +  $N$  octets)
- b) Payload ( $X$  octets)
- c) Trailer (2 octets)

[Note] In the following,  $N$  is the **Prefix** field length and  $X$  is the **Payload** field length.

[注] 以下では、 $N$  は Prefix フィールド長であり、 $X$  は Payload フィールド長である。

The structure of a **SpW-R Packet** is shown in Figure 4-2.

SpW-R Packet の構造を Figure 4-2 に示す。

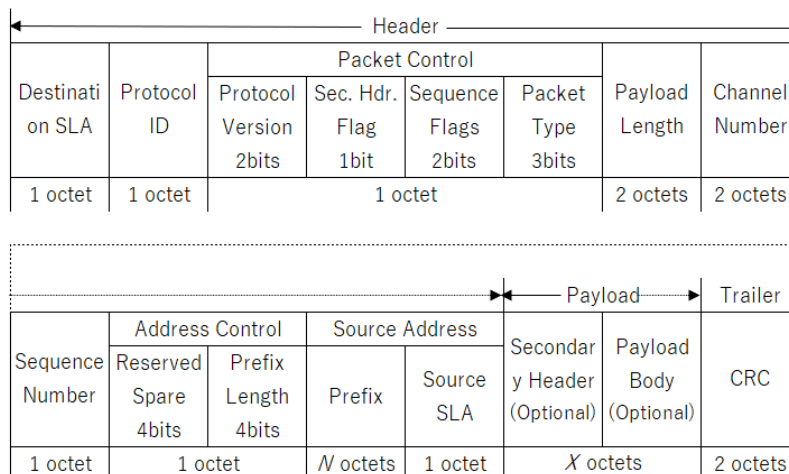


Figure 4-2: Structure of a SpW-R Packet

SpW-R Packet の構造

### 4.2.3. SPW-R PACKET HEADER

#### 4.2.3.1. GENERAL // 一般

The Header of a SpW-R Packet **shall** consist of the following fields, placed contiguously in the following order:

- a) Destination SLA (1 octet),
- b) Protocol ID (1 octet),
- c) Packet Control (1 octet),
- d) Payload Length (2 octets),
- e) Channel Number (2 octets),
- f) Sequence Number (1 octet),
- g) Address Control (1 octet), and
- h) Source Address ( $N+1$  octets).

For all types of Packets specified in Section 4.2.1, the same Header **shall** be used.

The value in the Source Address field of Data, Control, Flow Control, or Heartbeat Packet **shall** be copied into the destination identifiers and the Destination SLA field of the corresponding Ack Packet. In addition, the consistency of the values in between the Source Address field and the Channel Number field **may** be verified for the Packet of any type.

SpW-R Packet の Header は、以下の順序で連続して配置された、以下のフィールドからなること。

- a) Destination SLA (1 octet)
- b) Protocol ID (1 octet)
- c) Packet Control (1 octet)
- d) Payload Length (2 octets)
- e) Channel Number (2 octets)
- f) Sequence Number (1 octet)
- g) Address Control (1 octet)
- h) Source Address ( $N+1$  octets)

4.2.1 項が定める Packets の全ての種類が同じ Header を用いること。

Data, Control, Flow Control,または Heartbeat Packet の Source Address フィールドの値を、対応する Ack Packet の destination identifiers 及び Destination SLA フィールドにコピーすること。これに加え、あらゆる種類の Packet について、Source Address フィールドの値と Channel Number フィールドの値との整合性を検証しても良い。

#### 4.2.3.2. Destination SLA

According to Section 5.2.1 of [A1], **octet 0** (the first **octet**) of the **Header** **shall** be the **Destination SLA**, which is the **SpaceWire Logical Address (SLA)** that identifies the **node** to which the **Packet** is delivered.

The **Destination SLA** of a forward and reverse direction **SpW-R Packet** **shall** be **SLA** of the **Receive TEP** and **Transmit TEP** (see Section 4.2), respectively.

[A1] の 5.2.1 項に従って、**Header** の **octet 0** (最初の **octet**) は、**Packet** の配信先の **node** を特定する **SpaceWire Logical Address (SLA)** である **Destination SLA** である**こと**。

順方向及び逆方向の **SpW-R Packet** の **Destination SLA** は、それぞれ、**受信 TEP** 及び**送信 TEP** の **SLA** (4.2 項参照) である**こと**。

#### 4.2.3.3. Protocol ID

As specified in Section 5.2.2 of [A1], **octet 1** (the second **octet**) of the **Header** **shall** be the **Protocol ID** of this protocol, which is assigned by the **European Cooperation for Space Standardization (ECSS)**.

[Note] The **Protocol ID** of this protocol is 05h [A4].

[A1] の 5.2.2 項が定めるように、**Header** の **octet 1** (二番目の **octet**) は、**European Cooperation for Space Standardization (ECSS)** によって割り当てられたものである本プロトコルの **Protocol ID** である**こと**。

[注] 本プロトコルの **Protocol ID** は 05h である [A4]。

#### 4.2.3.4. Packet Control

##### 4.2.3.4.1. GENERAL // 一般

Octet 2 of the Header **shall** be the Packet Control field.

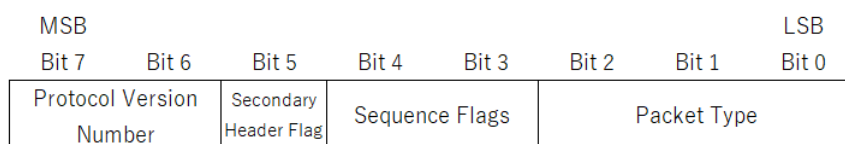
Header の octet 2 は、Packet Control フィールドであること。

The Packet Control field **shall** consist of the following sub-fields, placed contiguously in the following order (see Figure 4-3):

Packet Control フィールドは、以下の順序で連続して配置された、以下のサブフィールドからなること (Figure 4-3 参照)。

- a) Protocol Version Number (2 bits),
- b) Secondary Header Flag (1 bit),
- c) Sequence Flags (2 bits), and
- d) Packet Type (3 bits).

- A) Protocol Version Number (2 ビット)
- b) Secondary Header Flag (1 ビット)
- c) Sequence Flags (2 ビット)
- d) Packet Type (3 ビット)



**Figure 4-3: Structure of the Packet Control field**  
**Packet Control** フィールドの構造

##### 4.2.3.4.2. Protocol Version Number

Bits 6-7 of the Packet Control field **shall** be the Protocol Version Number of the SpW-R protocol specification with which the format of the Packet complies. These bits **shall** be set to “01” for this version of specification.

Packet Control フィールドの bits 6-7 は、Packet のフォーマットが準拠する SpW-R プロトコル仕様の Protocol Version Number であること。同ビットは、本バージョンの仕様では、“01” に設定すること。

##### 4.2.3.4.3. Secondary Header Flag

Bit 5 of the Packet Control field **shall** be the Secondary Header Flag. Value “1” of the flag **shall** indicate the presence of a variable-length Secondary Header (see Section 4.2.4.2) in the Payload field of the Packet. Value “0” **shall** indicate the absence of a Secondary Header.

Packet Control フィールドの bit 5 は、Secondary Header Flag であること。本フラグの値 “1” は、Packet の Payload フィールドに可変長である Secondary Header (4.2.4.2 項参照) が存在する事を示すこと。値 “0” は、Secondary Header が存在しない事を示すこと。

#### 4.2.3.4.4. Sequence Flags

Bits 3-4 of the Packet Control field **shall** be the Sequence Flags. Packet Control フィールドの bits 3-4 は、Sequence Flags であること。

The Sequence Flags field of a Data Packet **shall** indicate whether the Data Packet contains a complete service data unit or the first, intermediate, or final segment generated from a service data unit. The permitted values for the Sequence Flags field are shown in Table 4-1. Data Packet の Sequence Flags フィールドは、その Data Packet が、service data unit の全体か、service data unit から生成された最初、中間、または最後の segment か、の何れを含むかを示すこと。Sequence Flags フィールドに許される値を Table 4-1 に示す。

The other types of Packets than Data Packets **shall** have their Sequence Flags field set to “11”, indicating that they are stand-alone Packets. Data Packets 以外の種類の Packets では、単独 Packets である事を示すために、Sequence Flags フィールドを “11” に設定すること。

Table 4-1: Permitted Values for the Sequence Flags field  
Sequence Flags フィールドに許される値

Bit 4	Bit 3	Interpretation
0	1	Containing the first segment of a service data unit Service data unit の最初の segment を含む
0	0	Containing an intermediate segment of a service data unit Service data unit の中間 segment を含む
1	0	Containing the last segment of a service data unit Service data unit の最後の segment を含む
1	1	Containing one complete service data unit (stand-alone Packet) Service data unit の全体を含む (単独 Packet)

#### 4.2.3.4.5. Packet Type

Bits 0-2 of the Packet Control field shall be the Packet Type, which identifies the type of the Packet. Packet Control フィールドの bits 0-2 は、本 Packet の種別を特定するものである Packet Type であること。

The permitted values for the Packet Type are shown in Table 4-2. Packet Type に許される値を Table 4-2 に示す。

Table 4-2: Permitted Values for the Packet Type  
Packet Type に許される値

Value	Packet Type
0	Data Packet
1	Data Ack Packet
2	Control Packet (Open Command)
3	Control Packet (Close Command)
4	Heartbeat Packet
5	Heartbeat Ack Packet
6	Flow Control Packet or Flow Control Ack Packet (see Note)
7	Control Ack Packet

[Note] The Packet of Packet Type 6 that is sent from the Receive TEP to the Transmit TEP of a Transport Channel is interpreted as a Flow Control Packet, and the Packet of Packet Type 6 that is sent from the Transmit TEP to the Receive TEP is interpreted as a Flow Control Ack Packet.

[注] Transport Channel の受信 TEP から送信 TEP に送る Packet Type が 6 の Packet は、Flow Control Packet として解釈される。また、送信 TEP から受信 TEP に送る Packet Type が 6 の Packet は、Flow Control Ack Packet として解釈される。



#### 4.2.3.5. Payload Length

Octets 3-4 of the Header **shall** be the Payload Length, which specifies the number of the octets in the Payload field. The number counts any octets in neither the Header nor Trailer but does those in the Secondary Header if it is present (see Section 4.2.4.2).

Header の octets 3-4 は、Payload フィールドの octets 数を指定するものである Payload Length であること。その octets 数に、Header も Trailer の何れの octets も含めないが、Secondary Header が存在する場合 (4.2.4.2 項を参照) はその octets を含める。

#### 4.2.3.6. Channel Number

Octets 5-6 of the Header **shall** be the Channel Number, which specifies the Transport Channel to which the Packet belongs.

Header の octets 5-6 は、当 Packet が属する Transport Channel を指定するものである Channel Number であること。

Channel Numbers **shall** be uniquely assigned to Transport Channels in a SpaceWire network and **shall** be specified in a project-specific document using the table provided in Section 5.2 of this document.

一つの SpaceWire network において、Channel Numbers は Transport Channels に一意に割り当てられること。また、本書 5.2 項に掲載する表を用いてプロジェクト固有の文書に定めること。

#### 4.2.3.7. Sequence Number

Octet 7 of the Header **shall** be the Sequence Number, which identifies the location modulo 256 of the Packet in a series of Packets transferred through a Transport Channel.

For the Data Packet, the Sequence Number is assigned according to the procedure specified in Section 4.4.4.4.

For the Control Packet, the value of the Sequence Number is 0 (see Section 4.4.4.4).

For the Flow Control Packet, the Sequence Number is assigned according to the procedure specified in Section 4.5.3.7.

For the Heartbeat Packet, the value of the Sequence Number is 0 (see Sections 4.4.4.4 and 4.5.3.7).

For the Ack Packet, the Sequence Number is assigned according to the procedure specified in Section 4.5.3.4 (Data or Control Ack Packet), Section 4.4.4.6 (Flow Control Ack Packet), Section 4.4.4.7 (Received Heartbeat Ack Packet), and Section 4.5.3.6 (Transmit Heartbeat Ack Packet). Note that the Sequence Number of a Data, Control, Flow Control, or Heartbeat Packet is copied into that of the Ack Packet for the Packet.

[Note] If a Packet is retransmitted, its Sequence Number is the same as that in the first transmission (see Sections 4.4.4.5 and 4.5.3.7).

Header の octet 7 は、ある Transport Channel を介して伝送する一連の Packets において、当 Packet の位置を 256 のモジュロにより特定するものである Sequence Number であること。

Data Packet では、4.4.4.4 項が定める procedure に従って、Sequence Number を割り当てる。

Control Packet では、Sequence Number の値は 0 である (4.4.4.4 項参照)。

Flow Control Packet では、Sequence Number は 4.5.3.7 項が定める procedure に従って割り当てる。

Heartbeat Packet では、Sequence Number の値は 0 である (4.4.4.4 項及び 4.5.3.7 項参照)。

Ack Packet では、Sequence Number は 4.5.3.4 項 (Data または Control Ack Packet)、4.4.4.6 項 (Flow Control Ack Packet)、4.4.4.7 項 (Received Heartbeat Ack Packet) 、及び 4.5.3.6 項 (Transmit Heartbeat Ack Packet) が定める procedure に従って割り当てる。ここで、Data, Control, Flow Control, または Heartbeat Packet の Sequence Number は、その Packet に対応する Ack Packet の Sequence Number にコピーされる。

[注] Packet を再送する場合、その Packet の Sequence Number は、はじめの送信の際のものと同じである (4.4.4.5 項及び 4.5.3.7 項参照)。

#### 4.2.3.8. Address Control

##### 4.2.3.8.1. GENERAL // 一般

Octet 8 of the Header **shall** be the Address Control field.

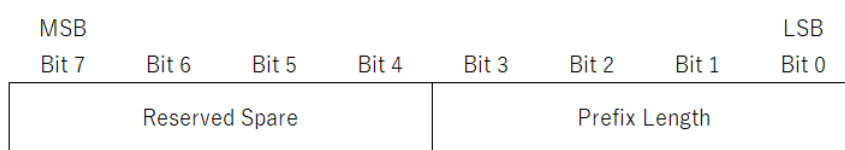
The Address Control field **shall** consist of the following sub-fields, placed contiguously in the following order (see Figure 4-4):

- a) Reserved Spare (4 bits), and
- b) Prefix Length (4 bits).

Header の octet 8 は、Address Control フィールドであること。

Address Control フィールドは、以下の順序で連続して配置された、以下のサブフィールドからなること (Figure 4-4 参照)。

- a) Reserved Spare (4 ビット)
- b) Prefix Length (4 ビット)



**Figure 4-4: Structure of the Address Control field**  
**Address Control**フィールドの構造

##### 4.2.3.8.2. Reserved Spare

Bits 4-7 of the Address Control field **shall** be the Reserved Spare, which is reserved for potential future use and **shall** be set to “0000”.

Address Control フィールドの bits 4-7 は、将来の潜在的な使用のために予約されたものである Reserved Spare であること。また、“0000” に設定すること。

##### 4.2.3.8.3. Prefix Length

Bits 0-3 of the Address Control field **shall** be the Prefix Length ( $N$ ), which specifies whether the Prefix field (see Section 4.2.3.9.2) exists or not and specifies the number of the octets that are contained in the Prefix field if the Prefix field exists.

Address Control フィールドの bits 0-3 は、Prefix Length ( $N$ ) であること。これは、Prefix フィールド (4.2.3.9.2 項参照) が存在するか否か、また Prefix フィールドが存在する場合 Prefix フィールドが含む octets の数を指定するものである。

[Note] In the case where only logical addressing is used in a SpaceWire network, the Prefix Length ( $N$ ) is set to zero in the SpaceWire network.

[注] ある SpaceWire network で logical addressing のみを用いる場合、その SpaceWire network では Prefix Length ( $N$ ) はゼロに設定する。

#### 4.2.3.9. Source Address

##### 4.2.3.9.1. GENERAL // 一般

The  $N+1$  octets beginning with octet 9 of the Header **shall** be the Source Address field.

The Source Address field of the Packet specifies a full variable-length source address, that identifies the node from which the Packet is transmitted.

The Source Address field **shall** consist of the following sub-fields, placed contiguously in the following order:

- a) Prefix ( $N$  octets), and
- b) Source SLA (1 octet).

The number of the octets ( $N$ ) of the Prefix field is specified in the Prefix Length field in the Address Control field (see Section 4.2.3.8.3).

[Note] In the case where only logical addressing is used in a SpaceWire network, the Prefix field is absent in the SpaceWire network.

Header の octet 9 から始まる  $N+1$  octets は、Source Address フィールドであること。

Packet の Source Address フィールドは、その Packet の送信元の node を特定するものである可変長の送信元アドレス全体を指定する。

Source Address フィールドは、以下の順序で連続して配置された、以下のサブフィールドからなること。

- a) Prefix ( $N$  octets)
- b) Source SLA (1 octet)

Prefix フィールドの octets の数 ( $N$ ) は、Address Control フィールドの Prefix Length フィールドで指定する (4.2.3.8.3 項参照)。

[注] ある SpaceWire network で logical addressing のみを用いる場合、その SpaceWire network では Prefix フィールドは存在しない。

#### 4.2.3.9.2. Prefix

If  $N$  is greater than zero, the first  $N$  octets of the Source Address field **shall** be the Prefix field, which consist of destination identifiers. If not, the Prefix field **shall** be absent.

The Source SLA of a forward and reverse direction SpW-R Packet **shall** be destination identifiers of the Transmit TEP and Receive TEP (see Section 4.2), respectively.

$N$  がゼロより大きい場合、Source Address フィールドの最初の  $N$  octets は、destination identifiers からなるものである Prefix フィールドであること。さもなければ、Prefix フィールドは存在しないこと。

順方向及び逆方向の SpW-R Packet の Source SLA は、それぞれ、送信 TEP 及び受信 TEP の destination identifiers (4.2 項参照) であること。

#### 4.2.3.9.3. Source SLA

The last octet of the Source Address field **shall** be the Source SLA field, which specifies the Source SpaceWire Logical Address (SLA).

The Source SLA of a forward and reverse direction SpW-R Packet **shall** be SLA of the Transmit TEP and Receive TEP (see Section 4.2), respectively

Source Address フィールドの最後の octet は、Source SpaceWire Logical Address (SLA) を指定するものである Source SLA フィールドであること。

順方向及び逆方向の SpW-R Packet の Source SLA は、それぞれ、送信 TEP 及び受信 TEP の SLA (4.2 項参照) であること。

## 4.2.4. SPW-R PACKET PAYLOAD

### 4.2.4.1. GENERAL // 一般

The **Payload** field of a SpW-R Packet **shall** consist of the following sub-fields, placed contiguously, in the following order:

- a) Secondary Header (optional), and
- b) Payload Body (optional).

The Secondary Header is present in the **Payload** field if Secondary Headers are used by the project. Otherwise, no Secondary Header is present (see Section 4.2.4.2).

The **Payload Body** field is as follows, depending on the type of the **Packet** specified in Section 4.2.1.

- a) For a **Data Packet**, the **Payload Body** field consists of only the Application Data (see Section 4.2.4.3).
- b) For a **Flow Control Packet**, as well as for **Data Ack** or **Control Ack Packet** if **Flow Control** is used in the **Transport Channel**, the **Payload Body** field consists of only the Maximum Acceptable Sequence Number (MASN) field (see Section 4.2.4.4).
- c) For the other types of **Packet** (*i.e.*, **Control**, **Heartbeat**, **Flow Control Ack**, or **Heartbeat Ack Packet** as well as for **Data Ack** or **Control Ack Packet** if **Flow Control** is not used in the **Transport Channel**), no **Payload Body** field is present.

SpW-R Packet の **Payload** フィールドは、以下の順序で連続して配置された、以下のサブフィールドからなること。

- a) Secondary Header (オプション)
- b) Payload Body (オプション)

プロジェクトで Secondary Headers を用いる場合 **Payload** フィールドに、Secondary Header は存在する。さもなければ、Secondary Header は存在しない(4.2.4.2 項参照)。

**Payload Body** フィールドは、4.2.1.項が定める **Packet** の種類に応じて以下のようにする。

- a) **Data Packet** では、**Payload Body** フィールドは、Application Data のみからなる (4.2.4.3 項参照)。
- b) **Flow Control Packet** において、並びに、**Transport Channel** に **Flow Control** を用いる時は **Data Ack** または **Control Ack Packet** において、**Payload Body** フィールドは、Maximum Acceptable Sequence Number (MASN) フィールドのみからなる (4.2.4.4 項参照)。
- c) その他 **Packet** では (つまり、**Control**, **Heartbeat**, **Flow Control Ack**, または **Heartbeat Ack Packet** において、並びに、**Transport Channel** に **Flow Control** を用いない時は **Data Ack** または **Control Ack Packet** において)、**Payload Body** フィールドは、存在しない。

#### 4.2.4.2. Secondary Header

If **Secondary Headers** are used by the project, the **Payload** field **shall** contain a field referred to as a **Secondary Header**. The **Secondary Header**, if present, **shall** begin with **octet 0** (the first **octet**) of the **Payload**.

The contents, length, and format of the **Secondary Header** **shall** be determined by each project and **shall** be documented in a project-specific document.

The presence of the **Secondary Header** is indicated by the **Secondary Header Flag** in the **Packet Control** field (see Section 4.2.3.4.3).

[Note] The **Secondary Header** can be used either for exchanging project-specific management information between the **Transmit and Receive TEPs** and/or for sending project-specific special data provided by the application.

#### 4.2.4.3. Application Data

The **Payload** field of a **Data Packet** **shall** contain a field referred to as an **Application Data** field. In terms of the position in a **Packet**, the **Application Data** field **shall** follow the **Secondary Header** if it is present or else **shall** follow the **Header**.

The **Application Data** field **shall** consist of a **segment** of a **service data unit** or a complete **service data unit**.

The maximum length of the **Application Data** field **shall** be determined for each **Transport Channel** and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

プロジェクトで **Secondary Headers** を用いる場合、**Payload** フィールドは、**Secondary Header** と称するフィールドを含むこと。**Secondary Header** は、存在する場合、**Payload** の **octet 0** (最初の **octet**) から始まること。

**Secondary Header** の中身、長さ、及び形式は、各プロジェクトが決めること。また、プロジェクト固有の文書に文書化すること。

**Secondary Header** の存在は、**Packet Control** フィールドの **Secondary Header Flag** によって示される (4.2.3.4.3 項を参照)。

[注] **Secondary Header** は、送信及び受信 TEPs の間でプロジェクト固有の管理情報を交換するためか、アプリケーションが提供するプロジェクト固有の特別なデータを送るためか、の何れか一方か双方かに使用できる。

**Data Packet** の **Payload** フィールドは、**Application Data** と称するフィールドを含むこと。**Packet** 内の位置としては、**Application Data** フィールドは、**Secondary Header** が存在する場合は **Secondary Header** に続くこと。さもないければ、**Header** に続くこと。

**Application Data** フィールドは、**service data unit** の **segment** か **service data unit** 全体かからなること。

**Application Data** フィールドの最大長は、**Transport Channel** 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

#### 4.2.4.4. Maximum Acceptable Sequence Number (MASN)

The maximum value of the Sequence Number that the Receive TEP can accept at a certain timing is referred to as the Maximum Acceptable Sequence Number (MASN). A MASN field **shall** take the value of the MASN.

The Payload field of a Flow Control Packet **shall** contain a MASN field. If Flow Control is used in the Transport Channel, the Payload field of a Data Ack or Control Ack Packet also **shall** contain a MASN field. In terms of the position in a Packet, the MASN field **shall** follow the Secondary Header if it is present or else **shall** follow the Header. The length of the MASN field **shall** be one octet.

受信 TEP がある時点で受理できる Sequence Number の最大値を、Maximum Acceptable Sequence Number (MASN) と称する。MASN フィールドは、MASN の値を取る**こと**。

Flow Control Packet の Payload フィールドは、MASN フィールドを含む**こと**。また、Transport Channel に Flow Control を用いる場合 Data Ack または Control Ack Packet の Payload フィールドも MASN フィールドを含む**こと**。Packet 内の位置としては、MASN フィールドは、Secondary Header が存在する場合は Secondary Header に続く**こと**。さもないければ、Header に続く**こと**。MASN フィールドの長さは 1 octet である**こと**。



#### 4.2.5. SPW-R PACKET TRAILER

The Trailer shall be a 16-bit CCITT CRC.

Trailer は、16 ビットの CCITT CRC であること。

The CRC shall be computed with regard to the octets from the Destination SLA to the last octet of the Payload contained in the Packet.

CRC は、Packet が含む Destination SLA から Payload の最後の octet までの octets に関して計算すること。

The CRC bits shall be computed according to the following polynomial:

CRC ビットは、以下の多項式に従って計算すること。

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

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

All the initial values for the computation shall be set to 1 before computing the CRC of each Packet.

各 Packet の CRC を計算する前に、計算の全ての初期値を 1 に設定すること。

### 4.3. STRUCTURE OF PROCEDURES // PROCEDURES の構造

The procedures conducted by the protocol entities プロトコル構成要素によって実行される procedures are classified into the following three categories: は、以下の三つのカテゴリに分類される。

- |  |                             |
|--|-----------------------------|
| a) <u>Transmit TEP Procedures</u> ,    | a) <u>送信 TEP Procedures</u> |
| b) <u>Receive TEP Procedures</u> , and | b) <u>受信 TEP Procedures</u> |
| c) <u>Node Procedures</u> .            | c) <u>Node Procedures</u>   |

Figure 4-5 shows the structure of the procedures used to implement SpaceWire-R at a SpaceWire node. Figure 4-5 は、SpaceWire node で SpaceWire-R を実装するために用いる procedures の構造を示す。

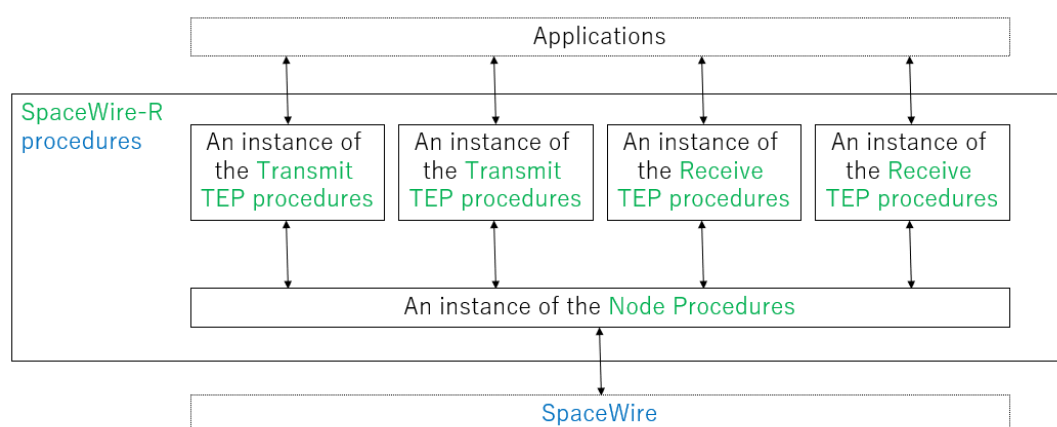


Figure 4-5: Structure of the Procedures at a Node  
Node の Procedures の構造

An instance of the set of Transmit TEP Procedures (specified in Section 4.4) shall exist for each of the Transmit TEPs that exist on the node. An instance of the set of Receive TEP Procedures (specified in Section 4.5) shall exist for each of the Receive TEPs that exists on the node. An instance of the set of Node Procedures (specified in Section 4.6) shall exist at each node.

ある node に存在する送信 TEPs の各々に、(4.4 項が定める) 送信 TEP Procedures 一式の instance が存在すること。ある Node に存在する受信 TEPs の各々に、(4.5 項が定める) 受信 TEP Procedures 一式の instance が存在すること。各 node に、(4.6 項が定める) Node Procedures 一式の instance が存在すること。



## 4.4. TRANSMIT TEP PROCEDURES // 送信 TEP PROCEDURES

### 4.4.1. GENERAL // 一般

This section specifies the **procedures** to be conducted by the **Transmit TEP**.

[Note] The **procedures** specified in this subsection are presented in an abstract sense and do not describe any specific ways of implementation of a **protocol entity**.

The **Transmit TEP** conducts the following three **procedures**:

- a) **Segmentation** (Segmenting),
- b) **Transmission Control**, and
- c) **Channel Control**.

These **procedures** and the logical relationship among them are shown in Figure 4-6.

The **Transmission Control procedure** has sub-**procedures** (see Section 4.4.4). Table 4-3 summarizes which (sub-)procedures are responsible for each processing step of each type of **Packet**.

本項は、**送信 TEP** が実行する **procedures** を定める。

[注] 本項が定める **procedures** は、抽象的な意味で提示されており、**プロトコル構成要素**の具体的な実装方法を記すものではない。

**送信 TEP** は、以下の三つの **procedures** を実行する。

- a) **Segmentation** (Segmenting)
- b) **送信制御**
- c) **Channel 制御**

これらの **procedures** と、それらの間の論理関係とを Figure 4-6 に示す。

**送信制御 procedure** は、サブ **procedure** を持つ (4.4.4 項参照)。また、Table 4-3 に、各種類の **Packet** の各処理ステップを何れの (サブ) **procedures** が担うかをまとめた。

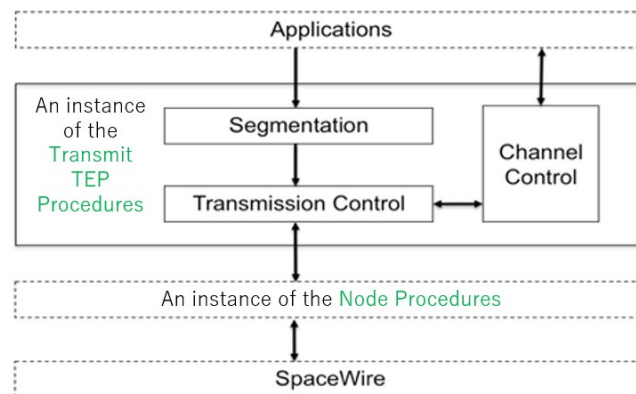


Figure 4-6: **Transmit TEP Procedures**  
**送信 TEP Procedures**

In what follows, **Flow Control Packets** are processed only if **Flow Control** is used. **Heartbeat Packets** are processed only if **Heartbeat** is used.

以下では、**Flow Control Packets** は、**Flow Control** を用いる場合にのみ処理する。**Heartbeat Packets** は、**Heartbeat** を用いる場合にのみ処理する。

Table 4-3: Summary of the **Transmit TEP Procedures** and **Receive TEP Procedures**

送信 TEP Procedures と受信 TEP Procedures のサマリ

	Step	Forward Direction			Reverse Direction		
		Data Packet	Control Packet	(transmit) Heartbeat Packet	(receive) Heartbeat Packet	Flow Control Packet	
Transmission Steps	Service	4.4.3 Channel Control		NA	NA	NA	
	generate	4.4.2 Segmentation		4.4.4.7 (1) (transmit) Heartbeat	4.5.3.6 (2) (receive) Heartbeat	4.5.3.5 Flow Control	
	generate 2ndary Header	4.4.4.8 Generation and Processing of Secondary Header			4.5.3.8 Processing and Generation of Secondary Header		
	transmit* <sup>1</sup>	4.4.4.4 Non-ACK Packet Transmission			4.5.3.7 Non-ACK Packet Transmission and Retransmission		
	retransmit	4.4.4.5 Non-ACK Packet Retransmission					
Reception Steps	receive	4.5.3.2 Packet Reception			4.4.4.2 Packet Reception		
	process 2ndary Header	4.5.3.8 Processing and Generation of Secondary Header			4.4.4.8 Generation and Processing of Secondary Header		
	transmit* <sup>1</sup> Ack Packet	4.5.3.4 Data and Control Packet Acceptance		4.5.3.6 (1) (transmit) Heartbeat	4.4.4.7 (2) (receive) Heartbeat	4.4.4.6 Flow Control	
	process	4.5.4 Reconstruction	4.5.2 Channel Control	NA	NA		
	Service					NA	

\*1: Includes the setting of the Sequence Number // Sequence Number の設定を含む

#### 4.4.2. SEGMENTATION (SEGMENTING)

While the **Transmit TEP** is in the **OPEN** state, it conducts the **procedure** specified in this section (see Section 4.4.3.4).

The **Transmit TEP** segments, according to this **procedure**, **service data units** that are provided by the sending application and are to be transmitted to the receiving application through the **TransferData.request primitive** (see Section 3.3.3.2).

One **instance** of this **procedure** **shall** exist for each **Transport Channel**.

If the length of a **service data unit** exceeds the maximum length of a **service data unit** (see Section 3.3.2.1), the **Transmit TEP** **shall** reject the transmission of the **service data unit** and **shall** notify the sending application about it through the **TransferData.Notify primitive** with the **Notification Type** “**Reject Transfer**” and the **Reason** “**SDU too long**” (see Section 3.3.3.3). If not, the **Transmit TEP** **shall** accept the transmission of the **service data unit**, **shall** notify the sending application about it through the **TransferData.Notify primitive** with the **Notification Type** “**Accept Transfer**” (see Section 3.3.3.3), and **shall** execute the following **procedure**.

If the length of a **service data unit** exceeds the maximum length of the **Application Data** field (see Section 4.2.4.3), the **Transmit TEP** **shall** divide the **service data unit** into **segments** each of which can fit in the **Application Data** field. Then, the **Transmit TEP** **shall** generate **Data Packets**, the total number of which is the same as that of the **segments**, and **shall** insert the **segments** into the **Application Data** fields of the **Data Packets**, setting an appropriate value to the **Sequence Flags** field (see Section 4.2.3.4.4) of each **Data Packet**.

If not, the **Transmit TEP** **shall** generate a **Data Packet** and **shall** insert the **service data unit** into the **Application Data** field of the **Data Packet**, setting an appropriate value to the **Sequence Flags** field.

In a **Transport Channel**, the **Data Packets** containing the **segments** that have been generated from a single **service data unit** **shall** be transmitted consecutively without being interlaced with any other **Data Packets** that carry (a **segment** of) another **service data unit**.

**送信 TEP** は、**OPEN** 状態にある間、本項が定める **procedure** を実行する (4.4.3.4 項参照)。

送信アプリケーションが **TransferData.request primitive** (3.3.3.2 項参照) を介して受信アプリケーションに送信するために提供した **service data units** を、**送信 TEP** は、本 **procedure** に従って分割する。

本 **procedure** の **instance** は、**Transport Channel** 毎に一つ存在すること。

**Service data unit** の長さが、**service data unit** の最大長 (3.3.2.1 項参照) を超える場合、**送信 TEP** は、その **service data unit** の送信を拒否し、その旨を、**Notification Type** が “**Reject Transfer**” かつ **Reason** が “**SDU too long**” の **TransferData.Notify primitive** (3.3.3.3 項参照) を介して送信アプリケーションに通知すること。さもなければ、**送信 TEP** は、その **service data unit** の送信を受理し、その旨を、**Notification Type** が “**Accept Transfer**” の **TransferData.Notify primitive** (3.3.3.3 項参照) を介して送信アプリケーションに通知し、以下の **procedure** を実行すること。

**Service data unit** の長さが、**Application Data** フィールドの最大長 (4.2.4.3 項参照) を超える場合、**送信 TEP** は、その **service data unit** を、その各々が **Application Data** フィールドに収まる **segments** に分割すること。次に、**送信 TEP** は、その **segments** と同数の **Data Packets** を生成し、それらの **Data Packets** の **Application Data** フィールドにその **segments** を挿入し、各 **Data Packet** の **Sequence Flags** フィールド (4.2.3.4.4 項参照) に適切な値を設定すること。

さもなければ、**送信 TEP** は、**Data Packet** を生成し、その **service data unit** をその **Data Packet** の **Application Data** フィールドに挿入し、**Sequence Flags** フィールドに適切な値を設定すること。

ある **Transport Channel** で、一つの **service data unit** から生成した **segments** を含む **Data Packets** は、他の **service data unit** (の **segment**) を運ぶ他の **Data Packets** と織り交ぜる事なく、連続して送信されること。



### 4.4.3. CHANNEL CONTROL // CHANNEL 制御

#### 4.4.3.1. GENERAL // 一般

The **Transmit TEP** controls its state machine according to this **procedure**.

本 **procedure** に従って、**送信 TEP** は、そのステートマシンを制御する。

One **instance** of this **procedure** **shall** exist for each **Transport Channel**.

本 **procedure** の **instance** は、**Transport Channel** 毎に一つ存在すること。

Each **Transmit TEP** **shall** be allowed to have the following four states:

各**送信 TEP** は、以下の四つの状態を持つ事ができること。

- a) **CLOSED**,
- b) **ENABLED**,
- c) **OPEN**, and
- d) **CLOSING**.

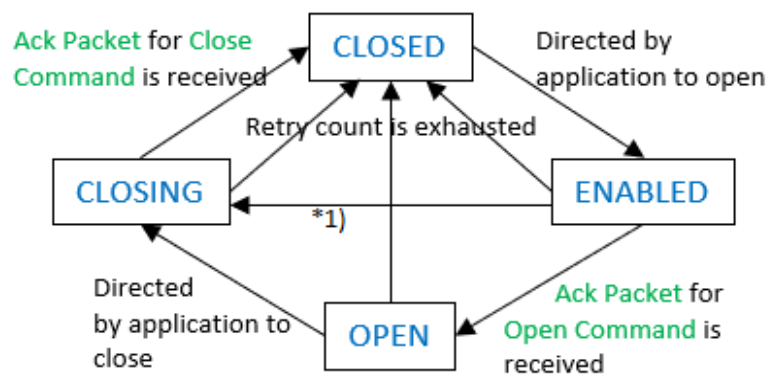
- a) **CLOSED**
- b) **ENABLED**
- c) **OPEN**
- d) **CLOSING**

Upon the occurrence of a state transition, the **Transmit TEP** **shall** notify the sending application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the occurrence of the state transition.

状態遷移が発生したら、**送信 TEP** は、状態遷移の発生を、**ChannelControl.indication primitive** (3.2.3.3 項参照) を介して送信アプリケーションに通知すること。

A summary of the state machine is shown in Figure 4-7.

ステートマシンの概要を Figure 4-7 に示す。



\*1) Directed by application to close and **Ack Packet for Open Command** is received

Figure 4-7: State Machine at the **Transmit TEP**  
**Transmit TEP** のステートマシン



#### 4.4.3.2. CLOSED

This section specifies the **procedure** which is conducted by the **Transmit TEP** in the **CLOSED** state.

Upon initialization, the **Transmit TEP** **shall** be in the **CLOSED** state.

The **Transmit TEP** in this state **shall** not generate any **Packets** and **shall** not respond to any received **Packets**.

Once the **Transmit TEP** in this state has received a directive from the sending application through the **ChannelControl.request primitive**, if the **Directive Type** is “Open” (see Section 3.2.3.2), the **Transmit TEP** **shall** transition into the **ENABLED** state, and if not, the **Transmit TEP** **shall** simply discard the directive.

If the **Transmit TEP** in this state has received a request to transmit a **service data unit** from the sending application through the **TransferData.request primitive** (see Section 3.3.3.2), the **Transmit TEP** **shall** reject the request and **shall** notify the sending application about the rejection through the **TransferData.indication primitive** with the **Notification Type** “Reject Transfer” and the **Reason** “Channel Not Open” (see Section 3.3.3.3).

本項は、**CLOSED** 状態にある**送信 TEP** が実行する **procedure** を定める。

**送信 TEP** は、初期化の後、**CLOSED** 状態にあること。

この状態にある**送信 TEP** は、**Packets** を生成しないこと。また、受信した **Packets** に応答しないこと。

この状態にある**送信 TEP** は、送信アプリケーションから **ChannelControl.request primitive** を介して指示を受けたら、その **Directive Type** が “Open” (3.2.3.2 項参照)であれば **ENABLED** 状態に移行すること。さもなければ単にその指示を破棄すること。

この状態にある**送信 TEP** は、**service data unit** を送信する要求を **TransferData.request primitive** (3.3.3.2 項参照) を介して送信アプリケーションから受けた場合、その要求を拒否し、拒否について、**Notification Type** が “Reject Transfer” かつ **Reason** が “Channel Not Open” の **TransferData.indication primitive** (3.3.3.3 項参照) を介して送信アプリケーションに通知すること。

#### 4.4.3.3. ENABLED

This section specifies the **procedure** which is conducted by the **Transmit TEP** in the **ENABLED** state.

Upon transitioning into this state, the **Transmit TEP** **shall** generate an **Open Command**. Then, the **Transmit TEP** transmits the **Open Command** and waits for a **Control Ack Packet** for the transmitted **Open Command** according to the **Transmission Control procedure** (see Section 4.4.4). Once the **Transmit TEP** has received the **Ack Packet** for the **Open Command**, the **Transmit TEP** **shall** transition into the **OPEN** state. Alternatively, if the retry count for retransmissions of the **Open Command** is exhausted, the **Transmit TEP** transitions into the **CLOSED** state (see Section 4.4.4.5).

The **Transmit TEP** in this state **shall** not generate any other **Packets** and **shall** not respond to any other received **Packets**.

Upon the reception of a request from the sending application through the **TransferData.request primitive** (see Section 3.3.3.2) to transmit a **service data unit**, the **Transmit TEP** in this state **shall** reject the request and **shall** notify the sending application about the rejection through the **TransferData.indication primitive** with the **Notification Type** “**Reject Transfer**” and the **Reason** “**Channel Not Open**” (see Section 3.3.3.3).

Once the **Transmit TEP** in this state has received a directive from the sending application through the **ChannelControl.request primitive**, if the **Directive Type** is “**Close**” (see Section 3.2.3.2), the **Transmit TEP** waits until it receives the **Ack Packet** for the transmitted **Open Command** (see Section 4.4.4.5) and then **shall** transition into the **CLOSING** state, and if not, the **Transmit TEP** **shall** simply discard the directive.

本項は、**ENABLED** 状態にある**送信 TEP** が実行する **procedure** を定める。

**送信 TEP** は、この状態へ移行時に、**Open Command** を生成すること。そして、**送信制御 procedure** に従って、**Open Command** を送信すると共に送信した **Open Command** に対する **Control Ack Packet** を待つ (4.4.4 項参照)。**送信 TEP** は、**Open Command** に対する **Ack Packet** を受信したら、**OPEN** 状態に移行すること。あるいは、**Open Command** の再送信の再試行回数が尽きた場合、**送信 TEP** は **CLOSED** 状態に移行する (4.4.4.5 項参照)。

この状態にある**送信 TEP** は、他の **Packets** を生成しないこと。また、受信した他の **Packets** に応答しないこと。

この状態にある**送信 TEP** は、**service data unit** を送信する要求を、**TransferData.request primitive** (3.3.3.2 項参照) を介して送信アプリケーションから受けたら、その要求を拒否し、拒否について、**Notification Type** が “**Reject Transfer**” かつ **Reason** が “**Channel Not Open**” の **TransferData.indication primitive** (3.3.3.3 項参照) を介して送信アプリケーションに通知すること。

この状態にある**送信 TEP** は、送信アプリケーションから **ChannelControl.request primitive** を介して指示を受けたら、その **Directive Type** が “**Close**” (3.2.3.2 項参照) であれば送信した **Open Command** に対する **Ack Packet** を受信するまで待機 (4.4.4.5 項参照) してから、**CLOSING** 状態に移行すること。さもなければ単にその指示を破棄すること。

#### 4.4.3.4. OPEN

This section specifies the **procedure** which is conducted by the **Transmit TEP** in the **OPEN** state.

Upon the reception of a request from the sending application through the **TransferData.request primitive** (see Section 3.3.3.2) to transmit a **service data unit**, the **Transmit TEP** in this state **shall** conduct, first, the **Segmentation procedure** (see Section 4.4.2). In the **procedure** the **Transmit TEP** generates **Data Packets**. Then, the **Transmit TEP** transmits the generated **Data Packets** and waits for the corresponding **Data Ack Packets**, both according to the **Transmission Control procedure** (see Section 4.4.4). If **tFlow Control** and/or **Heartbeat** are used in the **Transport Channel**, the **Transmit TEP shall** also conduct **Flow Control** and/or **Heartbeat**, respectively, according to the **Transmission Control procedure** (see Sections 4.4.4.6 and/or 4.4.4.7, respectively).

If the retry count for retransmissions of a **Data or Heartbeat Packet** is exhausted, the **Transmit TEP** transitions into the **CLOSED** state (see Section 4.4.4.5).

Once the **Transmit TEP** in this state has received a directive from the sending application through the **ChannelControl.request primitive**, if the **Directive Type** is “Close” (see Section 3.2.3.2), the **Transmit TEP shall** transition into the **CLOSING** state, and if not, the **Transmit TEP shall** simply discard the directive.

本項は、**OPEN** 状態にある**送信 TEP** が実行する **procedure** を定める。

この状態にある**送信 TEP** は、**service data unit** を送信する要求を、**TransferData.request primitive** (3.3.3.2 項参照) を介して送信アプリケーションから受けたら、まず **Segmentation procedure** (4.4.2 を参照) を実行すること。同 **procedure** では、**送信 TEP** は、**Data Packets** を生成する。次いで、**送信制御 procedure** (4.4.4 項参照) に従って生成された **Data Packets** を**受信 TEP** に送信し、対応する **Data Ack Packets** を待つ。また、**Transport Channel** に **Flow Control** 及び/または **Heartbeat** を用いる場合、**送信 TEP** は、**送信制御 procedure** に従って、それぞれ、**Flow Control** 及び/または **Heartbeat** を実行すること (それぞれ、4.4.4.6 項及び/または 4.4.4.7 項参照)。

**Data または Heartbeat Packet** の再送信の再試行回数が尽きた場合、**送信 TEP** は **CLOSED** 状態に移行する (4.4.4.5 項参照)。

この状態にある**送信 TEP** は、送信アプリケーションから **ChannelControl.request primitive** を介して指示を受けたら、その **Directive Type** が “Close” (3.2.3.2 項参照) であれば **CLOSING** 状態に移行すること。さもないければ単にその指示を破棄すること。

#### 4.4.3.5. CLOSING

This section specifies the **procedure** which is conducted by the **Transmit TEP** in the **CLOSING** state.

Upon transitioning into this state, the **Transmit TEP** **shall** generate a **Close Command**. Then, the **Transmit TEP** transmits the **Close Command** and waits for a **Control Ack Packet** for the transmitted **Close Command** according to the **Transmission Control procedure** (see Section 4.4.4). Once the **Transmit TEP** has received the **Ack Packet** for the **Close Command**, the **Transmit TEP** **shall** transition into the **CLOSED** state. Alternatively, if the retry count for retransmissions of the **Close Command** is exhausted, the **Transmit TEP** transitions into the **CLOSED** state (see Section 4.4.4.5).

The **Transmit TEP** in this state **shall** not generate any other **Packets** and **shall** not respond to any other received **Packets**.

If the **Transmit TEP** in this state receives a request from the sending application through the **TransferData.request primitive** (see Section 3.3.3.2) to transmit a **service data unit**, the **Transmit TEP** **shall** reject the request and **shall** notify the sending application about the rejection, through the **TransferData.indication primitive** with the **Notification Type** “**Reject Transfer**” and the **Reason** “**Channel Not Open**” (see Section 3.3.3.3).

If the **Transmit TEP** in this state receives a directive from the sending application through the **ChannelControl.request primitive**, the **Transmit TEP** **shall** discard the directive.

本項は、**CLOSING** 状態にある**送信 TEP** が実行する **procedure** を定める。

**送信 TEP** は、この状態への移行時に、**Close Command** を生成すること。そして、**送信制御 procedure** に従って、**受信 TEP** に **Close Command** を送信すると共に送信した **Close Command** に対する **Control Ack Packet** を待つ (4.4.4 項参照)。**送信 TEP** は、**Close Command** に対する **Ack Packet** を受信すると、**CLOSED** 状態に移行すること。あるいは、**Close Command** の再送信の再試行回数が尽きた場合、**送信 TEP** は **CLOSED** 状態に移行する (4.4.4.5 項参照)。

この状態にある**送信 TEP** は、他の **Packets** を生成しないこと。また、受信した他の **Packets** に応答しないこと。

この状態にある**送信 TEP** は、**service data unit** を送信する要求を、**TransferData.request primitive** (3.3.3.2 項参照) を介して送信アプリケーションから受けた場合、その要求を拒否し、拒否について、**Notification Type** が “**Reject Transfer**” かつ **Reason** が “**Channel Not Open**” の **TransferData.indication primitive** (3.3.3.3 項参照) を介して送信アプリケーションに通知すること。

この状態にある**送信 TEP** は、送信アプリケーションから **ChannelControl.request primitive** を介して指示を受けた場合、その指示を破棄すること。

#### 4.4.4. TRANSMISSION CONTROL // 送信制御

##### 4.4.4.1. General // 一般

The Transmit TEP performs the following according to this procedure.

- (1) The Transmit TEP receives Packets (see Section 4.4.4.2).
- (2) The Transmit TEP transmits the generated Data, Control, and (transmit) Heartbeat Packets to the Receive TEP and waits for the reception of the Ack Packets for the transmitted Packets (see Section 4.4.4.4). The Transmit TEP retransmits Data, Control, and (transmit) Heartbeat Packets if it has received no Ack Packets for the Packets (see Section 4.4.4.5).
- (3) The Transmit TEP transmits the Ack Packets for the Flow Control and (receive) Heartbeat Packets received from the Receive TEP (see Sections 4.4.4.6 and 4.4.4.7).
- (4) The Transmit TEP uses a sliding window (see Section 4.4.4.3) to manage the transmitting Data Packets.
- (5) Optionally, the Transmit TEP conducts Flow Control (see Section 4.4.4.6), Heartbeat (see Section 4.4.4.7) and/or Redundancy Control Support (see Section 4.4.4.9), in which the Transmit TEP processes the received Flow Control Packets and generates (transmit) Heartbeat Packets.

[Note] Data Packets are generated according to the Segmentation procedure (see Section 4.4.2). Control Packets are generated according to the Channel Control procedure (see Section 4.4.3).

One instance of this procedure shall exist for each Transport Channel.

送信 TEP は、本 procedure に従って、以下を実行する。

- (1) 送信 TEP は、Packets を受信する (4.4.4.2 項参照)。
- (2) 送信 TEP は、生成された Data, Control, 及び (送信) Heartbeat Packets を、受信 TEP に送信すると共にこれらの送信 Packets に対する Ack Packets の受信を待つ (4.4.4.4 項参照)。また、送信 TEP は、Data, Control, 及び (送信) Heartbeat Packets に対する Ack Packets を受信していない場合、これらの Packets を再送信する (4.4.4.5 項参照)。
- (3) 送信 TEP は、受信 TEP から受信した Flow Control 及び (受信) Heartbeat Packets に対する Ack Packets を送信する (4.4.4.6 項及び 4.4.4.7 項参照)。
- (4) 送信 TEP は、送信中の Data Packets を管理するために、sliding window (4.4.4.3 項参照) を用いる。
- (5) 送信 TEP は、オプションで、Flow Control (4.4.4.6 項参照), Heartbeat (4.4.4.7 項参照), 及び/または冗長系制御のサポート (4.4.4.9 項参照) を実行する。これらにより、送信 TEP は、受信した Flow Control Packets を処理し、また、(送信) Heartbeat Packets を生成する。

[注] Data Packets は、Segmentation procedure に従って生成される (4.4.2 項参照)。Control Packets は、Channel 制御 procedure に従って生成される (4.4.3 項参照)。

本 procedure の instance は、Transport Channel 毎に一つ存在すること。

#### 4.4.4.2. Packet Reception // Packet 受信

Incoming Packets are delivered to the Transmit TEP according to the Channel Demultiplexing procedure (see Section 4.6.3). The Transmit TEP determines whether it receives the incoming Packets or not according to this procedure.

The Transmit TEP **shall** examine each incoming Packet against a CRC error.

The Transmit TEP **shall** examine all the fields of the Header (see Figure 4-2) of all incoming Packets.

If a CRC error is detected for a Packet or if a Packet contains an unexpected value in any of the fields of the Header of a Packet, the Transmit TEP **shall** not receive the Packet. If not, the Transmit TEP **shall** receive the Packet.

If the Transmit TEP decides to receive the Packets, the received Packets are processed according to the procedures described in Section 4.4.4.6 (Flow Control Packets), Section 4.4.4.7 (2) (Heartbeat Packets), and Section 4.4.4.5 (Data Ack, Control Ack, and Heartbeat Ack Packets).

入力 Packets は、チャネル逆多重化 procedure に従って送信 TEP に配信される (4.6.3 項参照)。送信 TEP は、入力 Packets に対して、本 procedure に従って、それを受信するか拒否するかを決定する。

送信 TEP は、各入力 Packet について、CRC エラーに対する照合をすること。

送信 TEP は、全ての入力 Packets の Header の全てのフィールド (Figure 4-2 参照) を検査すること。

ある Packet に CRC エラーが検出された場合、あるいは、ある Packet が Header の何れかのフィールドに予期しない値を含む場合、送信 TEP は、その Packet は受信しないこと。そうでない限り、その Packet を受信すること。

もし送信 TEP がその Packets を受信すると決定した場合、同 Packets は、4.4.4.6 項 (Flow Control Packets)、4.4.4.7 (2) 項 (Heartbeat Packets)、及び 4.4.4.5 項 (Data Ack, Control Ack, 及び Heartbeat Ack Packets) が記す procedures に従って処理される。



#### 4.4.4.3. Sliding Window

A sliding window **shall** consist of  $k$  consecutive Sequence Numbers (modulo 256). Data Packets with Sequence Numbers within a sliding window might be “outstanding” (that is, they have been transmitted but the corresponding Ack Packets have not yet been received).

The size  $k$  of a sliding window **shall** be less than or equal to 128. It **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

Upon transitioning into the OPEN state (see Section 4.4.3.4), the Transmit TEP **shall** set the range of the Sequence Numbers within the sliding window to from 1 to  $k$  inclusive.

Let the range of the Sequence Numbers in the current sliding window be from  $n$  to  $n+k-1$ . Once the Transmit TEP has received the Ack Packet for a Data Packet with Sequence Number  $n$ , the Transmit TEP **shall** conduct either of the following processes; (1) if it has not received the Ack Packet for the Data Packet with Sequence Number  $n+1$ , the range of the Sequence Numbers within the sliding window **shall** be “slid” (aka shifted) by +1, i.e., from  $n+1$  to  $n+k$  inclusive, (2) if not, it means that the Transmit TEP has received by then the Ack Packets for the Data Packets with consecutive Sequence Numbers  $n+1$ , ...,  $n+p$ , for a certain positive integer  $p$ . In the latter case, the range of the Sequence Numbers within the sliding window **shall** be slid by  $p+1$ , i.e., from  $n+p+1$  to  $n+p+k$  inclusive.

Sliding window は、 $k$  個の連続する Sequence Numbers (モジュロ 256) からなること。Sliding window 内の Sequence Numbers を持つ Data Packets は、「応答未確認」である可能性がある（すなわち、送信したが、対応する Ack Packets をまだ受信していない）。

Sliding window のサイズ ( $k$ ) は、128 以下であること。このサイズは、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

送信 TEP は、OPEN 状態（4.4.3.4 項参照）への移行時に、sliding window が示す Sequence Numbers の範囲を 1 以上  $k$  以下に設定すること。

送信 TEP は、sliding window の範囲が  $n$  から  $n+k-1$  までのときに Sequence Number が  $n$  の Data Packet に対する Ack Packet を受信したら、以下の何れかの処理を実行すること。(1) Sequence Number が  $n+1$  の Data Packet に対する Ack Packet をまだ受信していない場合、sliding window が示す Sequence Numbers の範囲を +1 “スライド”させる（つまり、ずらす）、すなわち  $n+1$  から  $n+k$  までの範囲とすること。(2) さもなければ、Sequence Numbers がある正の整数  $p$  に対して  $n+1$ , ...,  $n+p$  と連続する Data Packets に対応する Ack Packet を送信 TEP がすでに受信している事を意味する。この後者の場合、sliding window が示す Sequence Numbers の範囲を  $p+1$  スライドさせる、すなわち  $n+p+1$  から  $n+p+k$  までの範囲とすること。

#### 4.4.4.4. Non-ACK Packet Transmission // 非 ACK Packet 送信

The Transmit TEP transmits and (when required and appropriate) retransmits Data and Control Packets, as well as Heartbeat Packets if Transmit Heartbeat is used, according to this procedure and the Non-ACK Packet Retransmission procedure (see Section 4.4.4.5), which are executed in the OPEN state for Data and Heartbeat Packets (see Section 4.4.3.4) and in the ENABLED and CLOSING states for Control Packets (see Sections 4.4.3.3 and 4.4.3.5, respectively).

The Transmit TEP shall transmit Data Packets generated by the Segmentation procedure (see Section 4.4.2), which contain a complete service data unit or its segments.

After the Transmit TEP transitions into the OPEN state, the Transmit TEP shall set the Sequence Number of the first Data Packet to 1.

Let  $m$  be the Sequence Number of the last transmitted Data Packet. The next Data Packet to be transmitted shall have Sequence Number  $m+1$ . If  $m+1$  is within the range of the current sliding window, the Data Packet shall be transmitted immediately. If not, its transmission shall be deferred until  $m$  increases to the point at which  $m+1$  satisfies the above-described condition. Note that if Flow Control is used, the fourth paragraph in Section 4.4.4.6 is applied instead of this paragraph.

Once the Transmit TEP has generated a Control Packet (see Section 4.4.3), the Transmit TEP shall transmit it immediately. The Sequence Number of the Control Packet shall be set to 0. After the Transmit TEP has transmitted the Control Packet, it shall transmit and retransmit no other Packets and shall not receive any incoming Packets (except for the Ack Packet for the transmitted Control Packet) until it receives the Ack Packet for the transmitted Control Packet.

Once the Transmit TEP has generated a Heartbeat Packet (see Section 4.4.4.7), the Transmit TEP shall transmit it immediately. The Sequence Number of the Heartbeat Packet shall be set to 0.

送信 TEP は、本 procedure と非 ACK Packet 再送信 procedure (4.4.4.5 項参照) に従って、Data 及び Control Packets を、またもし送信 Heartbeat を用いる場合は Heartbeat Packets も、送信、そして必要に応じて適宜再送信する。これらは、OPEN 状態で Data 及び Heartbeat Packets に対して (4.4.3.4 項参照)、ENABLED 状態及び CLOSING 状態で Control Packets に対して (それぞれ、4.4.3.3 項及び 4.4.3.5 項参照) 実行される。

送信 TEP は、Segmentation procedure が生成した Data Packets (4.4.2 項参照) を送信すること。これらの Data Packets には、service data unit の全体か、その segments が含まれる。

送信 TEP は、OPEN 状態に移行後、最初の Data Packet の Sequence Number を 1 に設定すること。

$m$  を一つ前に送信した Data Packet の Sequence Number とすると、次に送信する Data Packet の Sequence Number は  $m+1$  であること。 $m+1$  が現在の sliding window の範囲内である場合、その Data Packet は直ちに送信すること。さもなければ、その Data Packet の送信は、 $m+1$  が十分に増加して上記の条件を満たすまで延期すること。なお、Flow Control を用いる場合、この段落の代わりに 4.4.4.6 項の第 4 段落を適用する。

送信 TEP は、Control Packet を生成したら (4.4.3 項参照)、その Packet を直ちに送信すること。また、Control Packet の Sequence Number は、0 に設定すること。送信 TEP は、Control Packet を送信した後、他の Packets を送信も再送信もしないこと。また、送信した Control Packet に対する Ack Packet を受信するまでは、同 Ack Packet を除いたあらゆる入力 Packets を受信しないこと。

送信 TEP は、Heartbeat Packet を生成したら (4.4.4.7 項参照)、その Packet を直ちに送信すること。また、Heartbeat Packet の Sequence Number は、0 に設定すること。



#### 4.4.4.5. Non-ACK Packet Retransmission // 非 ACK Packet 再送信

Once the Transmit TEP has transmitted a Data, Control, or Heartbeat Packet according to the Non-ACK Packet Transmission procedure (see Section 4.4.4.4), the Transmit TEP **shall** generate and start a Transmit timer at the time when the last octet of the Packet is transmitted.

An independent Transmit timer **shall** be maintained for each outstanding Packet.

The initial value for the Transmit timers **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

Upon the reception of the Ack Packet for an outstanding Packet, the Transmit TEP **shall** discard the Transmit timer for the outstanding Packet.

Upon the reception of the Ack Packets for all the Data Packets associated with a service data unit, the Transmit TEP **shall** notify the sending application, through the DataTransferNotify.indication primitive with the Notification Type “Transfer Confirmed” (see Section 3.3.3.3), of the success of the transfer of the service data unit.

If the Transmit TEP has received no Ack Packet for a Packet within the period allowed by the Transmit timer for the Packet and if the retry count for the retransmission is less than or equal to the maximum retry count, the Transmit TEP **shall** retransmit the Packet with the original Sequence Number and **shall** restart the Transmit timer for the Packet.

The maximum retry count **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

送信 TEP は、非 ACK Packet 送信 procedure (4.4.4.4 項参照) に従って、Data, Control, または Heartbeat Packet を送信したら、Packet の最後の octet が送信されたタイミングで送信 timer を生成し、開始すること。

応答未確認の Packet 毎に、独立した送信 timer を維持すること。

送信 timers の初期値は、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

送信 TEP は、応答未確認の Packet に対する Ack Packet を受信したら、同応答未確認 Packet の送信 timer を破棄すること。

送信 TEP は、ある service data unit に関する全ての Data Packets に対する Ack Packets を受信したら、その service data unit の伝送の成功を、Notification Type が “Transfer Confirmed” の DataTransferNotify.indication primitive (3.3.3.3 項参照) を介して送信アプリケーションに通知すること。

送信 TEP が、ある Packet に対する Ack Packet を受信しないままその Packet の送信 timer が時間切れになり、かつ、再送信の再試行回数が最大再試行回数以下であれば、送信 TEP はその Packet を元の Sequence Number で再送信すると共にその Packet の送信 timer を再始動すること。

最大再試行回数は、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いて、プロジェクト固有の文書に定めること。

If the retry count for the retransmissions of a **Packet** exceeds the maximum retry count, the **Transmit TEP** **shall** declare the **Transport Channel** to be “inactive”, **shall** transition into the **CLOSED** state, and **shall** notify the sending application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the fact that the **Transport Channel** has been closed. If the failed **Packet** is a **Data Packet**, the **Transmit TEP** **shall** notify the sending application, through the **DataTransferNotify.indication primitive** with the **Notification Type** “Transfer Failure” (see Section 3.3.3.3), of the failure of the transfer of the **service data unit** from which the **Packet** has been generated.

If the **Transmit TEP** receives an **Ack Packet** for a non-outstanding **Packet** (*i.e.*, a **Packet** for which no **Transmit timer** is running), the **Transmit TEP** **shall** ignore it.

ある **Packet** の再送信の再試行回数が最大再試行回数を超えた場合、送信 TEP は、その **Transport Channel** は「非アクティブ」と宣言し、**CLOSED** 状態に移行すると共にその **Transport Channel** を閉じた事を、**ChannelControl.indication primitive** (3.2.3.3 項参照) を介して、送信アプリケーションに通知すること。また、失敗した **Packet** が **Data Packet** の場合、送信 TEP は、その **Packet** の生成元となる **service data unit** の伝送の失敗を、**Notification Type** が“Transfer Failure” の **DataTransferNotify.indication primitive** (3.3.3.3 項参照) を介して、送信アプリケーションに通知すること。

送信 TEP は、応答確認済の **Packet** (つまり、送信 timer が実行されていない **Packet**) の **Ack Packet** を受信した場合、その受信を無視すること。

#### 4.4.4.6. Flow Control

Whether or not Flow Control is used **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

While the Transmit TEP is in the OPEN state, it conducts the procedure specified in this section (see Section 4.4.3.4).

If Flow Control is used, the next paragraph **shall** substitute for the fourth paragraph in Section 4.4.4.4.

Let  $m$  be the Sequence Number of the last transmitted Data Packet. The next Data Packet to be transmitted **shall** have Sequence Number  $m+1$ . If  $m+1$  is within the range of the current sliding window and is less than or equal to the MASN contained in the most recent Ack or Flow Control Packet, the Data Packet **shall** be transmitted immediately. If not, its transmission **shall** be deferred until  $m$  increases to the point at which  $m+1$  satisfies the above-described condition.

Upon the reception of a Data Ack, Control Ack, or Flow Control Packet, the Transmit TEP **shall** record the MASN contained in the Packet.

Upon the reception of a Flow Control Packet, the Transmit TEP **shall** generate a Flow Control Ack Packet having the same Sequence Number as that of the received Flow Control Packet, and **shall** transmit the Flow Control Ack Packet. Note that the Transmit TEP transmits no Flow Control Ack Packet if the incoming Flow Control Packet has an error because the Flow Control Packet is not received by the Transmit TEP (see Section 4.4.4.2).

Flow Controlを用いるか否かは、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

送信 TEP は、OPEN 状態にある間、本項が定める procedure を実行する (4.4.3.4 項参照)。

Flow Controlを用いる場合、4.4.4.4 項の第 4 段落として次の段落を適用すること。

$m$  を一つ前に送信した Data Packet の Sequence Number とすると、次に送信する Data Packet の Sequence Number は  $m+1$  であること。 $m+1$  が現在の sliding window の範囲内であり、かつ、直近の Ack または Flow Control Packet が含む MASN 以下の場合、その Data Packet は直ちに送信すること。さもなければ、その Data Packet の送信は、 $m+1$  が十分に増加して上記の条件を満たすまで延期すること。

送信 TEP は、Data Ack, Control Ack, または Flow Control Packet を受信したら、その Packet が含む MASN を記憶すること。

送信 TEP は、Flow Control Packet を受信したら、受信した Flow Control Packet と同じ Sequence Number を持つ Flow Control Ack Packet を生成し、その Flow Control Ack Packet を送信すること。なお、入力 Flow Control Packet にエラーがある場合、送信 TEP は、その Flow Control Packet を受信しない (4.4.4.2 項参照) ため、Flow Control Ack Packet を送信しない。

#### 4.4.4.7. Heartbeat

Heartbeat can be used (1) for the Transmit TEP to check whether the link and the counterpart Receive TEP are still alive even when there are no data to transmit (Transmit Heartbeat), and (2) for the Receive TEP to check whether the link and the counterpart Transmit TEP are still alive even when there are no data to receive (Receive Heartbeat). It is possible to use both Transmit Heartbeat and Receive Heartbeat simultaneously on a Transport Channel. Whether or not Transmit Heartbeat is used and whether or not Receive Heartbeat is used **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

While the Transmit TEP is in the OPEN state, it conducts the procedure specified in this section (see Section 4.4.3.4).

##### (1) Transmit Heartbeat

If Transmit Heartbeat is used, a Transmit Heartbeat timer **shall** be employed by the Transmit TEP for each Transport Channel as follows.

The initial value of the Transmit Heartbeat timer **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

Upon transitioning into the OPEN state, the Transmit TEP **shall** start the Transmit Heartbeat timer. Whenever the Transmit TEP transmits a Packet (a Control, Data, or Ack Packet), the Transmit TEP **shall** restart the Transmit Heartbeat timer.

Once the Transmit Heartbeat timer has ended, the Transmit TEP **shall** generate a Heartbeat Packet and **shall** restart the Transmit Heartbeat timer. Then, the Transmit TEP transmits the Heartbeat Packet according to the Non-ACK Packet Transmission procedure (see Section 4.4.4.4).

Heartbeat は、(1) 送信 TEP が、送信するデータがない場合にも、link と対応する受信 TEP とがまだ生きているかどうかを確認するために (送信 Heartbeat)、また (2) 受信 TEP が、受信するデータがない場合にも、link と対応する送信 TEP とがまだ生きているかどうかを確認するために (受信 Heartbeat)、使用できる。ある Transport Channel で、送信 Heartbeat と受信 Heartbeat の双方を同時に使用できる。送信 Heartbeat を用いるか否か、並びに、受信 Heartbeat を用いるか否かは、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

送信 TEP は、OPEN 状態にある間、本項が定める procedure を実行する (4.4.3.4 項参照)。

##### (1) 送信 Heartbeat

送信 Heartbeat を用いる場合、各 Transport Channel の送信 TEP は、送信 Heartbeat timer を、以下のように用いること。

送信 Heartbeat timer の初期値は、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

送信 TEP は、OPEN 状態へ移行したら、送信 Heartbeat timerを開始すること。送信 TEP は、Packet (Control, Data, または Ack Packet) の送信のたびに、送信 Heartbeat timer を再始動すること。

送信 Heartbeat timer が時間切れになると、送信 TEP は、Heartbeat Packet を生成し、送信 Heartbeat timer を再始動すること。そして、送信 TEP は、非 ACK Packet 送信 procedure に従ってその Heartbeat Packet を送信する (4.4.4.4 項参照)。

## (2) Receive Heartbeat

If **Receive Heartbeat** is used, once the **Transmit TEP** has received a **Heartbeat Packet** from the **Receive TEP**, the **Transmit TEP** **shall** generate a **Heartbeat Ack Packet** that has the same **Sequence Number** as that of the received **Heartbeat Packet**, and **shall** transmit the **Heartbeat Ack Packet**. Note that the **Transmit TEP** transmits no **Heartbeat Ack Packet** if the incoming **Heartbeat Packet** has an error, because the **Heartbeat Packet** is not received by the **Transmit TEP** (see Section 4.4.4.2).

## (2) 受信 Heartbeat

**受信 Heartbeat** を用いる場合、**送信 TEP** は、**受信 TEP** から **Heartbeat Packet** を受信したら、受信した **Heartbeat Packet** と同じ **Sequence Number** を持つ **Heartbeat Ack Packet** を生成し、その **Heartbeat Ack Packet** を送信すること。なお、入力 **Heartbeat Packet** にエラーがある場合、**送信 TEP** は、その **Heartbeat Packet** を受信しない（4.4.4.2 項参照）ため、**Heartbeat Ack Packet** を送信しない。

#### 4.4.4.8. Generation and Processing of Secondary Header // Secondary Header の生成と処理

If the **Secondary Header** (see Section 4.2.4.2) is present, its contents **shall** be processed according to a **procedure** determined by each project, which **shall** be documented in a project-specific document.

**Secondary Header** (4.2.4.2 項参照) が存在する場合、その中身は、各プロジェクトが決めた **procedure** に従って処理すること。また、その処理を、プロジェクト固有の文書に文書化すること。

#### 4.4.4.9. Redundancy Control Support // 冗長系制御のサポート

[Note1] Once a **Transmit TEP** has declared the current **Transport Channel** to be “inactive” (see Section 4.4.4.5), the **Transmit TEP** notifies the sending application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the fact that the **Transport Channel** has been closed. In this case, the sending application might choose to open an alternative **Transport Channel**, using a separate route (in the same **SpaceWire network** or in another **SpaceWire network**) to the **receiving node**, and might retransmit all the outstanding **Data Packets** through the **Transport Channel**.

[注 1] **送信 TEP** は、現在の **Transport Channel** を「非アクティブ」であると宣言したら (4.4.4.5 項参照)、その **Transport Channel** を閉じた事を、**ChannelControl.indication primitive** (3.2.3.3 項参照) を介して送信アプリケーションに通知する。この場合、送信アプリケーションは、**受信 node** への別のルート (同じ **SpaceWire network** 内または別の **SpaceWire network** 内) を用いて代替 **Transport Channel** を開く事を選択し、その **Transport Channel** を介して全ての応答未確認の **Data Packets** を再送信する事がある。

[Note2] When a **Transmit TEP** declares the current **Transport Channel** to be “inactive”, the cause of the “inactive” mark might be that the delivery of the **Ack Packets** for **Data Packets** has failed although the **Data Packets** have been received correctly by the **Receive TEP**. In this case, the sending application is not informed of the fact that the **Data Packets** have been actually received by the **Receive TEP** and therefore will possibly retransmit the same **Data Packets** through a new **Transport Channel**. The receiving application must handle this type of duplicated retransmissions.

[注 2] **送信 TEP** が、現在の **Transport Channel** を「非アクティブ」であると宣言した時には、「非アクティブ」宣言の原因として、**受信 TEP** は **Data Packets** を正しく受信したが、それらの **Data Packets** に対する **Ack Packets** の配信に失敗したことがある。この場合、送信アプリケーションは、**受信 TEP** が **Data Packets** を実際に受信したという事実を知らされなくて、従って新しい **Transport Channel** を介して同じ **Data Packets** を再送信する可能性がある。受信アプリケーションは、この種の重複した再送信を処理する必要がある。



## 4.5. RECEIVE TEP PROCEDURES // 受信 TEP PROCEDURES

### 4.5.1. GENERAL // 一般

This section specifies the **procedures** to be conducted by the **Receive TEP**.

[Note] The **procedures** specified in this subsection are presented in an abstract sense and do not describe any specific ways of implementation of a **protocol entity**.

The **Receive TEP** conducts the following three **procedures**:

- a) **Channel Control**,
- b) **Reception Control**, and
- c) **Reconstruction** (Reassemble).

These **procedures** and the logical relationship among them are shown in Figure 4-8.

The **Reception Control procedure** has sub-**procedures** (see Section 4.4.4). Table 4-3 (see Section 4.4.1) summarizes which (sub-)procedures are responsible for each processing step of each type of **Packet**.

本項は、**受信 TEP** が実行する **procedures** を定める。

[注] 本項が定める **procedures** は、抽象的な意味で提示されており、**プロトコル構成要素**の具体的な実装方法を記すものではない。

**受信 TEP** は、以下の三つの **procedures** を実行する。

- a) **Channel 制御**
- b) **受信制御**
- c) **Reconstruction** (Reassemble)

これらの **procedures** と、それらの間の論理的な関係とを Figure 4-8 に示す。

**受信制御 procedure** は、サブ **procedure** を持つ(4.4.4 項参照)。また、Table 4-3 (4.4.1 項参照) に、各種類の **Packet** の各処理ステップを何れの(サブ) **procedures** が担うかをまとめた。

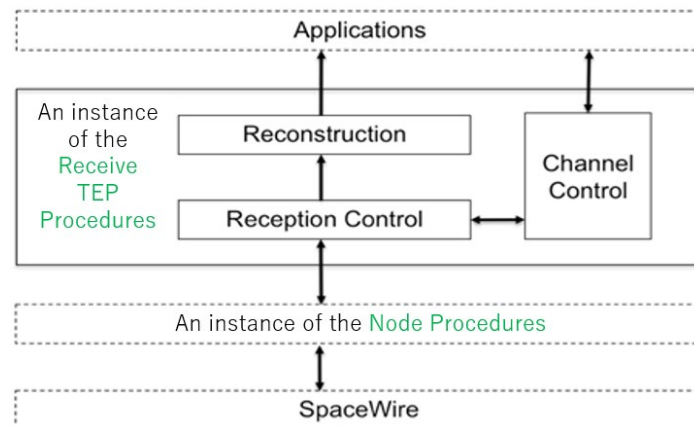


Figure 4-8: **Receive TEP Procedures**  
**受信 TEP Procedures**

In what follows, **Flow Control Packets** are processed only if **Flow Control** is used. **Heartbeat Packets** are processed only if **Heartbeat** is used.

以下では、**Flow Control Packets** は、**Flow Control** を用いる場合にのみ処理する。**Heartbeat Packets** は、**Heartbeat** を用いる場合にのみ処理する。





## 4.5.2. CHANNEL CONTROL // CHANNEL 制御

### 4.5.2.1. GENERAL // 一般

The **Receive TEP** controls its state machine according to this **procedure**.

One **instance** of this **procedure** **shall** exist for each **Transport Channel**.

Each **Receive TEP** **shall** be allowed to have the following four states.

- a) **CLOSED**,
- b) **ENABLED**,
- c) **OPEN**, and
- d) **CLOSING**.

Upon the occurrence of a state transition, the **Receive TEP** **shall** notify the receiving application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the occurrence of the state transition.

A summary of the state machine is shown in Figure 4-9.

本 **procedure** に従って、**受信 TEP** は、そのステートマシンを制御する。

本 **procedure** の **instance** は、**Transport Channel** 毎に一つ存在すること。

各**受信 TEP** は、以下の四つの状態を持つ事ができること。

- a) **CLOSED**
- b) **ENABLED**
- c) **OPEN**
- d) **CLOSING**

状態遷移が発生したら、**受信 TEP** は、状態遷移の発生を、**ChannelControl.indication primitive** (3.2.3.3 項参照)を介して受信アプリケーションに通知すること。

ステートマシンの概要を Figure 4-9 に示す。

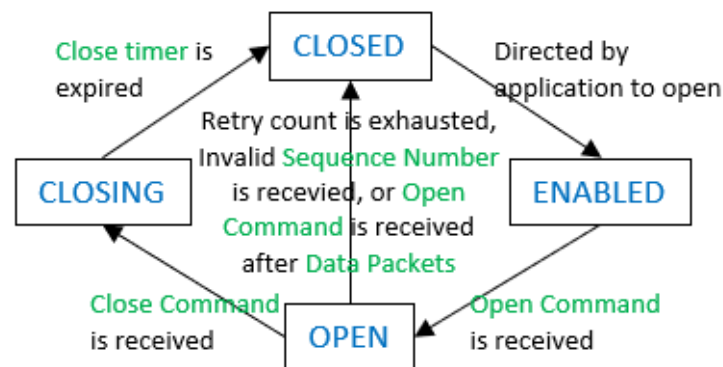


Figure 4-9: State Machine at the **Receive TEP**  
**Receive TEP** のステートマシン

#### 4.5.2.2. CLOSED

This section specifies the **procedure** which is conducted by the **Receive TEP** in the **CLOSED** state.

Upon initialization, the **Receive TEP** **shall** be in the **CLOSED** state.

The **Receive TEP** in this state **shall** not generate any **Packets** and **shall** not respond to any received **Packets**.

Once the **Receive TEP** in this state has received a directive from the receiving application through the **ChannelControl.request primitive**, if the **Directive Type** is “Open” (see Section 3.2.3.2), the **Receive TEP** **shall** transition into the **ENABLED** state, and if not, the **Receive TEP** **shall** simply discard the directive.

本項は、**CLOSED** 状態にある**受信 TEP** が実行する **procedure** を定める。

**受信 TEP** は、初期化の後、**CLOSED** 状態にあること。

この状態にある**受信 TEP** は、**Packets** を生成しないこと。また、受信した **Packets** に応答しないこと。

この状態にある**受信 TEP** は、受信アプリケーションから **ChannelControl.request primitive** を介して指示を受けたら、その **Directive Type** が “Open” (3.2.3.2 項参照) であれば **ENABLED** 状態に移行すること。さもなければ単にその指示を破棄すること。

### 4.5.2.3. ENABLED

This section specifies the **procedure** which is conducted by the **Receive TEP** in the **ENABLED** state.

The **Receive TEP** in this state waits for an **Open Command** and once the **Receive TEP** has received an **Open Command** with **Sequence Number** 0, the **Receive TEP** transmits the **Control Ack Packet** for the **Open Command**, according to the **Reception Control procedure** (see Section 4.5.3.4 (2) a)). Then, the **Receive TEP** **shall** transition into the **OPEN** state.

If the **Receive TEP** in this state receives a **Control Packet** whose **Sequence Number** is not 0, the **Receive TEP**, according to the **Reception Control procedure** (see Section 4.5.3.4 (2) d)), (1) does not accept the **Packet**, (2) does not transmit the **Ack Packet** for the **Packet**, (3) declares the **Transport Channel** to be “inactive”, and (4) transitions into the **CLOSED** state.

The **Receive TEP** in this state **shall** not generate any other **Packets** and **shall** not respond to any other received **Packets**.

If the **Receive TEP** in this state receives a directive from the receiving application through the **ChannelControl.request primitive**, the **Receive TEP** **shall** discard the directive.

本項は、**ENABLED** 状態にある**受信 TEP** が実行する **procedure** を定める。

この状態にある**受信 TEP** は、**Open Command** を待ち、**Sequence Number** が 0 の **Open Command** を受信したら、**受信制御 procedure** (4.5.3.4 (2) a)項参照) に従って、同 **Open Command** に対する **Control Ack Packet** を送信する。その後、**受信 TEP** は、**OPEN** 状態に移行すること。

この状態にある**受信 TEP** は、**Sequence Number** が 0 でない **Control Packet** を受信した場合、**受信制御 procedure** (4.5.3.4 (2) d)項参照) に従って、(1) その **Packet** を受理せず、(2) その **Packet** に対する **Ack Packet** を送信しない。また、(3) その **Transport Channel** は「非アクティブ」であると宣言し、(4) **CLOSED** 状態に移行する。

この状態にある**受信 TEP** は、他の **Packets** を生成しないこと。また、受信した他の **Packets** に応答しないこと。

この状態にある**受信 TEP** は、受信アプリケーションから **ChannelControl.request primitive** を介して指示を受けた場合、その指示は破棄すること。

#### 4.5.2.4. OPEN

This section specifies the **procedure** which is conducted by the **Receive TEP** in the **OPEN** state.

The **Receive TEP** in this state waits for **Data Packets**, and once the **Receive TEP** has received **Data Packets**, the **Receive TEP** transmits the **Data Ack Packets** for the **Data Packets** according to the **Reception Control procedure** (see Section 4.5.3.4 (1)). If **Flow Control** and/or **Heartbeat** are used in the **Transport Channel**, the **Receive TEP** **shall** also conduct **Flow Control** and/or **Heartbeat**, respectively, according to the **Reception Control procedure** (see Sections 4.5.3.5 and/or 4.5.3.6, respectively).

Upon the reception of an **Open Command**, the **Receive TEP** in this state conducts either of the following **procedures** according to the **Reception Control procedure** (see Section 4.5.3.4 (2) b) c) d)). If the **Receive TEP** has received any **Data Packets** since it transitioned into the **OPEN** state or if the **Sequence Number** of the received **Data Packet** is not 0, the **Receive TEP** does not accept the **Open Command**, declares the **Transport Channel** to be “inactive”, and transitions into the **CLOSED** state. If not, the **Receive TEP** does not reaccept the **Open Command** and transmits the **Control Ack Packet** for the **Open Command**.

If the **Receive TEP** in this state receives a **Data Packet** with an invalid **Sequence Number**, the **Receive TEP** declares the **Transport Channel** to be “inactive” and transitions into the **CLOSED** state (see Section 4.5.3.4 (1) d)).

If the retry count for retransmissions of a **Flow Control** or **Heartbeat Packet** is exhausted, the **Receive TEP** declares the **Transport Channel** to be “inactive” and transitions into the **CLOSED** state (see Section 4.5.3.7).

本項は、**OPEN** 状態にある**受信 TEP** が実行する **procedure** を定める。

この状態にある**受信 TEP** は、**Data Packets** を待ち、**Data Packets** を受信したら、**受信制御 procedure** に従ってそれらの **Data Packets** に対する **Data Ack Packets** を送信する (4.5.3.4 (1) 項参照)。また、ある **Transport Channel** に **Flow Control** 及び/または **Heartbeat** を用いる場合、**受信 TEP** は、**受信制御 procedure** に従って、それぞれ、**Flow Control** 及び/または **Heartbeat** を実行すること (それぞれ、4.5.3.5 項及び/または 4.5.3.6 項参照)。

この状態にある**受信 TEP** は、**Open Command** を受信したら、**受信制御 procedure** (4.5.3.4 (2) b) c) d)項参照) に従って、以下の何れかの **procedure** を実施する。もし自身が**OPEN** 状態に移行してからそれまでの間に **Data Packets** を受信していた場合、もしくは受信した **Data Packet** の **Sequence Number** が 0 でない場合、その **Open Command** は受理せず、その **Transport Channel** は「非アクティブ」とであると宣言し、**CLOSED** 状態に移行する。さもなければ、その **Open Command** は再受理せず、**Open Command** に対する **Control Ack Packet** は送信する。

この状態にある**受信 TEP** は、無効な **Sequence Number** の **Data Packet** を受信した場合、その **Transport Channel** は「非アクティブ」とであると宣言し、**CLOSED** 状態に移行する (4.5.3.4 (1) d)項参照)。

**Flow Control** または **Heartbeat Packet** の再送信の再試行回数が尽きた場合、**受信 TEP** は、その **Transport Channel** は「非アクティブ」とであると宣言し、**CLOSED** 状態に移行する (4.5.3.7 項参照)。

The Receive TEP in this state also waits for a Close Command. And once the Receive TEP has received a Close Command with Sequence Number 0, the Receive TEP transmits the Control Ack Packet for the Close Command, according to the Reception Control procedure (see Section 4.5.3.4 (2) a)). Then, the Receive TEP shall transition into the CLOSING state.

If the Receive TEP in this state receives a Close Command whose Sequence Number is not 0, the Receive TEP, according to the Reception Control procedure (see Section 4.5.3.4 (2) d)), (1) does not accept the Packet, (2) does not transmit the Ack Packet for the Packet, (3) declares the Transport Channel to be “inactive”, and (4) transitions into the CLOSED state.

If the Receive TEP in this state receives a directive from the receiving application through the ChannelControl.request primitive, the Receive TEP shall discard the directive.

この状態にある受信 TEP は、Close Command の受信を待ち、Sequence Number が 0 の Close Command を受信したら、受信制御 procedure (4.5.3.4 (2) a)項参照) に従ってその Close Command に対する Control Ack Packet を送信する。その後、受信 TEP は、CLOSING 状態に遷移すること。

この状態にある受信 TEP は、Sequence Number が 0 でない Close Command を受信した場合、受信制御 procedure (4.5.3.4 (2) d)項参照) に従って、(1) その Packet を受理せず、(2) その Packet に対する Ack Packet を送信しない。また、(3) その Transport Channel は「非アクティブ」とであると宣言し、(4) CLOSED 状態に移行する。

この状態にある受信 TEP は、受信アプリケーションから ChannelControl.request primitive を介して指示を受けた場合、その指示を破棄すること。

#### 4.5.2.5. CLOSING

This section specifies the **procedure** which is conducted by the **Receive TEP** in the **CLOSING** state.

Upon transitioning into this state, the **Receive TEP** **shall** start a **Close timer**. Then, once the **Close timer** has ended, the **Receive TEP** **shall** transition into the **CLOSED** state.

本項は、**CLOSING** 状態にある**受信 TEP** が実行する **procedure** を定める。

**受信 TEP** は、この状態への移行時に、**Close timer** を開始する**こと**。そして、**Close timer** が時間切れになると、**受信 TEP** は **CLOSED** 状態に移行する**こと**。

The initial value for the **Close timer** **shall** be determined for each **Transport Channel** as the maximum time for retransmitting the **Close Command**, which is greater than the product of the initial value of the **Transmit timer** and the maximum retry count for the retransmission (both see Section 4.4.4.5), and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

Once the **Receive TEP** in this state has received a **Close Command** with **Sequence Number** 0, the **Receive TEP** does not reaccept the **Close Command** and transmits the **Ack Packet** for the **Close Command** according to the **Reception Control procedure** (see Section 4.5.3.4 (2) b)).

If the **Receive TEP** in this state receives a **Control Packet** whose **Sequence Number** is not 0, the **Receive TEP**, according to the **Reception Control procedure** (see Section 4.5.3.4 (2) d)), (1) does not accept the **Packet**, (2) does not transmit the **Ack Packet** for the **Packet**, (3) declares the **Transport Channel** to be “inactive”, and (4) transitions into the **CLOSED** state.

The **Receive TEP** in this state **shall** not generate any other **Packets** and **shall** not respond to any other received **Packets**.

If the **Receive TEP** in this state receives a directive from the receiving application through the **ChannelControl.request primitive**, the **Receive TEP** **shall** discard the directive.

**Close timer** の初期値は、**送信 timers** の初期値と再送信の最大再試行回数 (共に 4.4.4.5 項参照) の積より大きな値である「**Close Command** の再送にかかる最大時間」として **Transport Channel** 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

この状態にある**受信 TEP** は、**Sequence Number** が 0 の **Close Command** を受信したら、その **Close Command** は再受理せず、**受信制御 procedure** (4.5.3.4 (2) b)項参照) に従って **Close Command** に対する **Ack Packet** を送信する。

この状態にある**受信 TEP** は、**Sequence Number** が 0 でない **Control Packet** を受信した場合、**受信制御 procedure** (4.5.3.4 (2) d)項参照) に従って、(1) その **Packet** を受理せず、(2) その **Packet** に対する **Ack Packet** を送信しない。また、(3) その **Transport Channel** は「**非アクティブ**」であると宣言し、(4) **CLOSED** 状態に移行する。

この状態にある**受信 TEP** は、他の **Packets** を生成しないこと。また、受信した他の **Packets** に応答しないこと。

この状態にある**受信 TEP** は、受信アプリケーションから **ChannelControl.request primitive** を介して指示を受けた場合、その指示は破棄すること。



## 4.5.3. RECEPTION CONTROL // 受信制御

### 4.5.3.1. General // 一般

The **Receive TEP** performs the following according to this **procedure**.

- (1) The **Receive TEP** receives **Packets** (see Section 4.5.3.2).
- (2) The **Receive TEP** transmits the **Ack Packets** for the **Data, Control, and (transmit) Heartbeat Packets** received from the **Transmit TEP** (see Sections 4.5.3.4 and 4.5.3.6).
- (3) The **Receive TEP** transmits the generated **Flow Control and (receive) Heartbeat Packets** to the **Transmit TEP** and waits for the reception of the **Ack Packets** for the transmitted **Packets** (see Section 4.5.3.7). The **Receive TEP** retransmits the **Flow Control and (receive) Heartbeat Packets** if it has received no **Ack Packets** for the **Packets** (see Section 4.5.3.7).
- (4) The **Receive TEP** uses a **sliding window** (see Section 4.5.3.3) to manage the receiving **Data Packets**. The **Receive TEP** accepts the received **Data and Control Packets** (see Section 4.5.3.4).
- (5) Optionally, the **Receive TEP** conducts **Flow Control** (see Section 4.5.3.5), **Heartbeat** (see Section 4.5.3.6), and/or **Redundancy Control Support** (see Section 4.5.3.9), in which the **Receive TEP** generates **Flow Control Packets** and (receive) **Heartbeat Packets**.

[Note] The accepted **Data Packets** are processed according to the **Reconstruction procedure** (see Section 4.5.4) and the accepted **Control Packets** are processed according to the **Channel Control procedure** (see Section 4.5.2).

One **instance** of this **procedure** **shall** exist for each **Transport Channel**.

本 **procedure** に従って、**受信 TEP** は、以下を実行する。

- (1) **受信 TEP** は、**Packets** を受信する (4.5.3.2 項参照)。
- (2) **受信 TEP** は、**送信 TEP** から受信した **Data, Control, 及び (送信) Heartbeat Packets** に対する **Ack Packets** を送信する (4.5.3.4 項及び 4.5.3.6 項参照)。
- (3) また、**受信 TEP** は、生成された **Flow Control 及び (受信) Heartbeat Packets** を、**送信 TEP** に送信すると共にこれらの送信 **Packets** に対する **Ack Packets** の受信を待つ (4.5.3.7 項参照)。また、**受信 TEP** は、**Flow Control 及び (受信) Heartbeat Packets** に対する **Ack Packets** を受信していない場合、これらの **Packets** を再送信する (4.5.3.7 項参照)。
- (4) **受信 TEP** は、受信中の **Data Packets** を管理するために、**sliding window** (4.5.3.3 項参照) を用いる。**受信 TEP** は、受信した **Data 及び Control Packets** を受理する (4.5.3.4 項参照)。
- (5) また、**受信 TEP** は、オプションで、**Flow Control** (4.5.3.5 項参照)、**Heartbeat** (4.5.3.6 項参照)、及び/または**冗長系制御のサポート** (4.5.3.9 項参照) を実行する。これらにより、**受信 TEP** は、**Flow Control Packets** 及び (受信) **Heartbeat Packets** を生成する。

[注] 受理された **Data Packets** は、**Reconstruction procedure** に従って処理される (4.5.4 項参照)。また受理された **Control Packets** は、**Channel 制御 procedure** に従って処理される (4.5.2 項参照)。

本 **procedure** の **instance** は、**Transport Channel** 毎に一つ存在すること。

#### 4.5.3.2. Packet Reception // Packet 受信

Incoming Packets are delivered to the Receive TEP according to the Channel Demultiplexing procedure (see Section 4.6.3). The Receive TEP determines whether it receives the incoming Packets or not according to this procedure.

The Receive TEP shall examine each incoming Packet against a CRC error.

The Receive TEP shall examine all the fields of the Header (see Figure 4-2) of all incoming Packets.

If a CRC error is detected for a Packet or if a Packet contains an unexpected value in any of the fields of the Header of a Packet, the Receive TEP shall not receive the Packet. If not, the Receive TEP shall receive the Packet.

If the Receive TEP decides to receive the Packets, the received Packets are processed according to the procedures described in Section 4.5.3.4 (1) (Data Packets), Section 4.5.3.4 (2) (Control Packets), Section 4.5.3.6 (1) (Heartbeat Packets), and Section 4.5.3.7 (Flow Control Ack and Heartbeat Ack Packets).

入力 Packets は、チャネル逆多重化 procedure に従って受信 TEP に配信される (4.6.3 項参照)。受信 TEP は、入力 Packets に対して、本 procedure に従って、それを受信するか拒否するかを決定する。

受信 TEP は、各入力 Packet について、CRC エラーに対する照合をすること。

受信 TEP は、全ての入力 Packets の Header の全てのフィールド (Figure 4-2 参照) を検査すること。

ある Packet に CRC エラーが検出された場合、あるいは、ある Packet が Header の何れかのフィールドに予期しない値を含む場合、受信 TEP は、その Packet は受信しないこと。そうでない限り、その Packet を受信すること。

もし受信 TEP がその Packets を受信すると決定した場合、同 Packets は、4.5.3.4 (1) 項 (Data Packets), 4.5.3.4 (2) 項 (Control Packets), 4.5.3.6 (1) 項 (Heartbeat Packets), 及び 4.5.3.7 項 (Flow Control Ack 及び Heartbeat Ack Packets) が記す procedures に従って処理される。

### 4.5.3.3. Sliding Window

A sliding window **shall** consist of  $k$  consecutive Sequence Numbers (modulo 256). The Receive TEP accepts Data Packets with Sequence Numbers within a sliding window (see Section 4.5.3.4).

The size  $k$  of a sliding window **shall** be the same as the size of the sliding window used at the counterpart Transmit TEP of the Transport Channel (see Section 4.4.4.3).

Upon transitioning into the OPEN state (see Section 4.5.2.4), the Receive TEP **shall** set the range of the Sequence Numbers within the sliding window to from 1 to  $k$  inclusive.

Let the range of the Sequence Numbers in the current sliding window be from  $n$  to  $n+k-1$ . Once the Receive TEP has received a Data Packet with Sequence Number  $n$ , the Receive TEP **shall** conduct either of the following processes; (1) if it has not received the Data Packet with Sequence Number  $n+1$ , the range of the Sequence Numbers within the sliding window **shall** be “slid” (aka shifted) by  $+1$ , i.e., from  $n+1$  to  $n+k$  inclusive, (2) if not, it means that the Receive TEP has received by then the Data Packets with consecutive Sequence Numbers  $n+1, \dots, n+p$ , for a certain positive integer  $p$ . In the latter case, the range of the Sequence Numbers within the sliding window **shall** be slid by  $p+1$ , i.e., from  $n+p+1$  to  $n+p+k$  inclusive.

Sliding window は、 $k$  個の連続する Sequence Numbers (モジュロ 256) からなること。受信 TEP は、sliding window 内の Sequence Numbers を持つ Data Packets を受取する (4.5.3.4 項参照)。

Sliding window のサイズ ( $k$ ) は、その Transport Channel の対応する送信 TEP で用いる sliding window のサイズと同じであること (4.4.4.3 項参照)。

受信 TEP は、OPEN 状態 (4.5.2.4 項参照) への移行時に、sliding window が示す Sequence Numbers の範囲を 1 以上  $k$  以下に設定すること。

受信 TEP は、sliding window の範囲が  $n$  から  $n+k-1$  までのときに Sequence Number が  $n$  の Data Packet を受信したら、以下の何れかの処理を実行すること。

- (1) Sequence Number が  $n+1$  の Data Packet をまだ受信していない場合、sliding window が示す Sequence Numbers の範囲を  $+1$  “スライド” させる (つまり、ずらす)、すなわち  $n+1$  から  $n+k$  までの範囲とすること。
- (2) さもないければ、Sequence Numbers がある正の整数  $p$  に対して  $n+1, \dots, n+p$  と連続する Data Packets を受信 TEP がすでに受信している事を意味する。この後者の場合、sliding window が示す Sequence Numbers の範囲を  $p+1$  スライドさせる、すなわち  $n+p+1$  から  $n+p+k$  までの範囲とすること。

#### 4.5.3.4. Data or Control Packet Acceptance // Data または Control Packet 受信

Upon the successful reception (i.e., without an error) of a **Control or Data Packet** (see Section 4.5.3.2), the **Receive TEP shall** conduct either of the following two **procedures**.

(1) The **Receive TEP shall** process the reception of a **Data Packet** as follows, the **procedure** of which depends on the **Sequence Number** of the **Packet**. Here, it is assumed that the **Sequence Numbers** within the current **sliding window** range from  $n$  to  $n+k-1$ . Note that if **Flow Control** is used in the **Transport Channel**, the two paragraphs at the end of Section 4.5.3.5 is applied instead of this paragraph and following itemized paragraphs a) and b).

- a) If the **Sequence Number** is in the current range of the **Sequence Numbers** within the **sliding window** and if the **Receive TEP** is in the **OPEN** state and has not accepted the **Packet** yet, the **Receive TEP shall** accept the **Packet**, **shall** generate the **Ack Packet** for the **Packet**, and **shall** transmit the **Ack Packet**.
- b) If the **Sequence Number** is in the current range of the **Sequence Numbers** within the **sliding window** and if the **Receive TEP** is in the **OPEN** state and has already accepted the **Packet**, the **Receive TEP shall** not reaccept the **Packet** but **shall** retransmit the **Ack Packet** for the **Packet** (because none of the former deliveries of the **Ack Packet** might have succeeded).
- c) If the **Sequence Number** is in the range from  $n-k$  to  $n-1$  inclusive (i.e. if there is a possibility that the **Sequence Number** is in the current range of the **sliding window** of the **Transmit TEP**) and if the **Receive TEP** is in the **OPEN** state, the **Receive TEP shall** not reaccept the **Packet** but **shall** retransmit the **Ack Packet** for the **Packet** (because although the **Packet** has already been accepted, none of the former deliveries of the **Ack Packet** for the **Packet** might have succeeded).

受信 TEP は、**Control または Data Packet** をエラー (4.5.3.2 項参照) なしで受信すると、以下の二つのうちの何れかの **procedures** を実施すること。

(1) 受信 TEP は、**Data Packet** の受信を、その **Packet** の **Sequence Number** に応じた **procedure** で、以下のように処理すること。ここで、現在の **sliding window** が示す **Sequence Numbers** の範囲は  $n$  から  $n+k-1$  であるとする。なお、**Transport Channel** に **Flow Control** を用いる場合、この段落と箇条書きされた以下の段落 a) 及び b) の代わりに 4.5.3.5 項の末尾の二つの段落を適用する。

- a) **Sequence Number** が **sliding window** が示す **Sequence Numbers** の現在範囲内であり、かつ、受信 TEP が **OPEN** 状態にあり、かつ、その **Packet** をまだ受理していない場合、受信 TEP は、その **Packet** を受理し、その **Packet** に対する **Ack Packet** を生成し、その **Ack Packet** を送信すること。
- b) **Sequence Number** が **sliding window** が示す **Sequence Numbers** の現在範囲内であり、かつ、受信 TEP が **OPEN** 状態にあり、かつ、その **Packet** をすでに受理している場合、受信 TEP は、その **Packet** は再受理しないが、その **Packet** に対する **Ack Packet** は再送信すること (その **Ack Packet** の以前の配信に一つも成功していない可能性があるため)。
- c) **Sequence Number** が  $n-k$  から  $n-1$  までの範囲にあり (つまり、**Sequence Number** が送信 TEP の **sliding window** が示す **Sequence Numbers** の現在範囲内である可能性あり)、かつ、受信 TEP が **OPEN** 状態にある場合、受信 TEP は、その **Packet** は再受理しないが、その **Packet** に対する **Ack Packet** は再送信すること (その **Packet** はすでに受理しているが、その **Packet** に対する **Ack Packet** の以前の配信に一つも成功していない可能性があるため)。

- d) If none of the above is the case, the **Receive TEP shall** not accept the **Packet** and **shall** not transmit the **Ack Packet** for the **Packet** (because the **Transmit TEP** is not functioning correctly). The **Receive TEP shall** declare the **Transport Channel** to be “inactive” and **shall** transition into the **CLOSED** state.
- (2) The **Receive TEP shall** process the reception of a **Control Packet** as follows.
- a) If the **Receive TEP** is in the **ENABLED** state and receives an **Open Command** with **Sequence Number 0** or if it is in the **OPEN** state and receives a **Close Command** with **Sequence Number 0**, the **Receive TEP shall** accept the **Command**, **shall** generate the **Ack Packet** for the **Command**, and **shall** transmit the **Ack Packet**. The **Receive TEP** starts a state transition according to the **Channel Control procedure** (see Sections 4.5.2.3 and 4.5.2.4)
- b) If the **Receive TEP** is in the **OPEN** state and receives an **Open Command** with **Sequence Number 0** before receiving any **Data Packets** or if the **Receive TEP** is in the **CLOSING** state and receives a **Close Command** with **Sequence Number 0**, the **Receive TEP shall** not reaccept the **Command** but **shall** retransmit the **Ack Packet** for the **Command** (because although the **Packet** has already been accepted, none of the former deliveries of the **Ack Packet** for the **Packet** might have succeeded).
- c) If the **Receive TEP** is in the **OPEN** state and receives an **Open Command** with **Sequence Number 0** after receiving one or more **Data Packets**, the **Receive TEP shall** not accept the **Command** and **shall** not transmit the **Ack Packet** for the **Command** (because the **Transmit TEP** is not functioning correctly). The **Receive TEP shall** declare the **Transport Channel** to be “inactive” and **shall** transition into the **CLOSED** state.
- d) 上記の条件のいずれも当てはまらない場合、受信 TEP は、その **Packet** を受取せず、その **Packet** に対する **Ack Packet** を送信しないこと（送信 TEP が正しく機能していないため）。また、その **Transport Channel** は「非アクティブ」とであると宣言し、**CLOSED** 状態に移行すること。
- (2) 受信 TEP は、**Control Packet** の受信を、以下のよう処理すること。
- a) 受信 TEP が **ENABLED** 状態で **Sequence Number** が 0 の **Open Command** を受信した場合、または **OPEN** 状態で **Sequence Number** が 0 の **Close Command** を受信した場合、受信 TEP は、その **Command** を受取し、その **Command** に対する **Ack Packet** を生成し、その **Ack Packet** を送信すること。受信 TEP は、**Channel 制御 procedure** (4.5.2.3 項及び 4.5.2.4 項参照) に従って状態遷移を開始する。
- b) 受信 TEP が **OPEN** 状態で **Data Packets** を受信する前に **Sequence Number** が 0 の **Open Command** を受信した場合、または **CLOSING** 状態で **Sequence Number** が 0 の **Close Command** を受信した場合、受信 TEP は、その **Command** は再受取しないが、その **Command** に対する **Ack Packet** は再送信すること（その **Packet** はすでに受取したが、その **Packet** に対する **Ack Packet** の以前の配信に一つも成功していない可能性があるため）。
- c) 受信 TEP が **OPEN** 状態で一つ以上の **Data Packets** を受信した後に **Sequence Number** が 0 である **Open Command** を受信した場合、受信 TEP は、その **Command** を受取せず、その **Command** に対する **Ack Packet** を送信しないこと（送信 TEP が正しく機能していないため）。また、その **Transport Channel** は「非アクティブ」とであると宣言し、**CLOSED** 状態に移行すること。

d) If the **Receive TEP** is not in the **CLOSED** state and receives a **Control Packet** with the **Sequence Number** of not 0, the **Receive TEP** **shall** not accept the **Packet** and **shall** not transmit the **Ack Packet** for the **Packet** (because the **Transmit TEP** is not functioning correctly), **shall** declare the **Transport Channel** to be “inactive”, and **shall** transition into the **CLOSED** state.

The **Receive TEP** **shall** copy the value of the **Sequence Number** of the received **Data** or **Control Packet** into the corresponding **Data Ack** or **Control Ack Packet**.

[Note] The **Receive TEP** transmits no **Ack Packet** if the incoming **Packet** has an error, because the **Packet** is not received by the **Receive TEP** (see Section 4.5.3.2).

When the **Receive TEP** transitions into the **CLOSED** state according to this **procedure**, the **Receive TEP** **shall** notify the receiving application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the fact that the **Transport Channel** has been closed.

d) **CLOSED** 状態にない受信 TEP が **Sequence Number** が 0 でない **Control Packet** を受信した場合、受信 TEP は、その **Packet** を受理せず、その **Packet** に対する **Ack Packet** を送信しないこと (送信 TEP が正しく機能していないため)。また、その **Transport Channel** は「非アクティブ」であると宣言し、**CLOSED** 状態に移行すること。

受信 TEP は、受信した **Data** または **Control Packet** の **Sequence Number** の値を、対応する **Data Ack** または **Control Ack Packet** にコピーすること。

[注] 入力 **Packet** にエラーがある場合、受信 TEP は、その **Packet** を受信しない (4.5.3.2 項参照) ため、**Ack Packet** を送信しない。

受信 TEP は、本 **procedure** に従って **CLOSED** 状態に移行時に、その **Transport Channel** を閉じた事を、**ChannelControl.indication primitive** (3.2.3.3 項参照) を介して受信アプリケーションに通知すること。



#### 4.5.3.5. Flow control

Whether or not Flow Control is used **shall** be determined for each Transport Channel and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

While the Receive TEP is in the OPEN state, it conducts the procedure specified in this section (see Section 4.5.2.4).

If Flow Control is used, a Flow Control, Data Ack, or Control Ack Packet carries MASN (see Section 4.2.4.4).

[Note 1] When a Transport Channel is opened, the initial value of the MASN is carried in the Control Ack Packet for the received Open Command.

[Note 2] Carrying a value of a MASN in the Control Ack Packet for a received Close Command would be meaningless. Therefore, the Control Ack Packet for a received Close Command **may** carry the default value of the MASN.

The MASN **shall** be within the current sliding window.

When the value of the MASN has changed and if there is no Data Ack Packet to be transmitted, the Receive TEP **shall** generate a Flow Control Packet carrying the new value of the MASN. Then, the Receive TEP transmits the Flow Control Packet according to the Non-ACK Packet Transmission procedure (see Section 4.5.3.7). After the Receive TEP has transmitted the Flow Control Packet, the Receive TEP might retransmit the Flow Control Packet (see Section 4.5.3.7). The Receive TEP **shall** generate no other Flow Control Packets until it receives the Ack Packet for the transmitted Flow Control Packet (see Section 4.5.3.7).

If Flow Control is used, the following paragraphs **shall** substitute for the paragraphs in Section 4.5.3.4 (1) a) and b).

Flow Control を用いるか否かは、Transport Channel 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

受信 TEP は、OPEN 状態にある間、本項が定める procedure を実行する (4.5.2.4 項参照)。

Flow Control を用いる場合、Flow Control, Data Ack, または Control Ack Packet は、MASN を運ぶ (4.2.4.4 項参照)。

[注 1] ある Transport Channel を開くと、受信した Open Command に対する Control Ack Packet が、MASN の初期値を運ぶ。

[注 2] 受信した Close Command に対する Control Ack Packet で、MASN の値を運ぶ事には意味がない。そこで、受信した Close Command に対する Control Ack Packet は、MASN のデフォルト値を運ぶと良い。

MASN は、現在の sliding window 内にあること。

受信 TEP は、送信する Data Ack Packet がない時には、MASN の値を変更すると、Packet 送信 procedure (4.5.3.7 項参照) に従って MASN の新しい値を運ぶ Flow Control Packet を生成すること。そして、受信 TEP は、非 ACK Packet 送信 procedure に従ってその Flow Control Packet を送信する (4.5.3.7 項参照)。受信 TEP は、Flow Control Packet を送信した後、その Flow Control Packet を再送信する可能性はある (4.5.3.7 項参照)。受信 TEP は、送信した Flow Control Packet に対する Ack Packet を受信するまで他の Flow Control Packets は生成しないこと。

Flow Control を用いる場合、4.5.3.4 (1) 項の段落 a) 及び b) として以下の段落を適用すること。

- a) If the Sequence Number is in the current range of the Sequence Numbers within the sliding window and is less than or equal to the MASN, and if the Receive TEP is in the OPEN state and has not accepted the Packet yet, the Receive TEP **shall** accept the Packet, **shall** generate the Ack Packet for the Packet, and **shall** transmit the Ack Packet.
- b) If the Sequence Number is in the current range of the Sequence Numbers within the sliding window and is less than or equal to the MASN, and if the Receive TEP is in the OPEN state and has already accepted the Packet, the Receive TEP **shall** not reaccept the Packet but **shall** retransmit the Ack Packet for the Packet (because none of the former deliveries of the Ack Packet might have succeeded).
- a) Sequence Number が sliding window が示す Sequence Numbers の現在範囲内かつ MASN 以下 であり、かつ、受信 TEP が OPEN 状態にあり、かつ、その Packet をまだ受理していない場合、受信 TEP は、その Packet を受理し、その Packet に対する Ack Packet を生成し、その Ack Packet を送信すること。
- b) Sequence Number が sliding window が示す Sequence Numbers の現在範囲内かつ MASN 以下 であり、かつ、受信 TEP が OPEN 状態にあり、かつ、その Packet をすでに受理している場合、受信 TEP は、その Packet は再受理しないが、その Packet に対する Ack Packet は再送信すること (その Ack Packet の以前の配信に一つも成功していない可能性があるため)。



#### 4.5.3.6. Heartbeat

(See the first paragraph in Section 4.4.4.7)

(4.4.4.7 項の第 1 段落を参照)

While the **Receive TEP** is in the **OPEN** state, it conducts the **procedure** specified in this section (see Section 4.5.2.4).

**受信 TEP** は、**OPEN** 状態にある間、本項が定める **procedure** を実行する (4.5.2.4 項参照)。

##### (1) Transmit Heartbeat

##### (1) 送信 Heartbeat

If **Transmit Heartbeat** is used, once the **Receive TEP** has received a **Heartbeat Packet** from the **Transmit TEP**, the **Receive TEP** **shall** generate a **Heartbeat Ack Packet** that has the same **Sequence Number** as that of the received **Heartbeat Packet**, and **shall** transmit the **Heartbeat Ack Packet**. Note that the **Receive TEP** transmits no **Heartbeat Ack Packet** if the incoming **Heartbeat Packet** has an error, because the **Heartbeat Packet** is not received by the **Receive TEP** (see Section 4.5.3.2).

**送信 Heartbeat** を用いる場合、**受信 TEP** は、**送信 TEP** から **Heartbeat Packet** を受信したら、受信した **Heartbeat Packet** と同じ **Sequence Number** を持つ **Heartbeat Ack Packet** を生成し、その **Heartbeat Ack Packet** を送信すること。なお、入力 **Heartbeat Packet** にエラーがある場合、**受信 TEP** は、その **Heartbeat Packet** を受信しない (4.5.3.2 項参照) ため、**Heartbeat Ack Packet** を送信しない。

##### (2) Receive Heartbeat

##### (2) 受信 Heartbeat

If **Receive Heartbeat** is used, a **Receive Heartbeat timer** **shall** be employed by the **Receive TEP** for each **Transport Channel** as follows.

**受信 Heartbeat** を用いる場合、各 **Transport Channel** の **受信 TEP** は、**受信 Heartbeat timer** を、以下のように用いること。

The initial value of the **Receive Heartbeat timer** **shall** be determined for each **Transport Channel** and **shall** be specified in a project-specific document, using the table provided in Section 5.2 of this document.

**受信 Heartbeat timer** の初期値は、**Transport Channel** 毎に決めること。また、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

Upon transitioning into the **OPEN** state, the **Receive TEP** **shall** start the **Receive Heartbeat timer**. Whenever the **Receive TEP** transmits a **Packet** (an **Ack** or **Flow Control Packet**), the **Receive TEP** **shall** restart the **Receive Heartbeat timer**.

**受信 TEP** は、**OPEN** 状態へ移行したら、**受信 Heartbeat timer** を開始すること。**受信 TEP** は、**Packet** (**Ack** または **Flow Control Packet**) の送信のたびに、**受信 Heartbeat timer** を再始動すること。

Once the **Receive Heartbeat timer** has ended, the **Receive TEP** **shall** generate a **Heartbeat Packet** and **shall** restart the **Receive Heartbeat timer**. Then, the **Receive TEP** transmits the **Heartbeat Packet** according to the **Non-ACK Packet Transmission procedure** (see Section 4.5.3.7).

**受信 Heartbeat timer** が時間切れになると、**受信 TEP** は、**Heartbeat Packet** を生成し、**受信 Heartbeat timer** を再始動すること。そして、**受信 TEP** は、**非 ACK Packet 送信 procedure** に従ってその **Heartbeat Packet** を送信する (4.5.3.7 項参照)。

#### 4.5.3.7. Non-ACK Packet Transmission and Retransmission // 非 ACK Packet 送信及び再送信

The Receive TEP transmits and (when required and appropriate) retransmits Flow Control Packets (if Flow Control is used in the Transport Channel) and/or Heartbeat Packets (if Receive Heartbeat is used in the Transport Channel) according to this procedure, which is executed in the OPEN state (see Section 4.5.2.4).

Once the Receive TEP has generated a Flow Control or Heartbeat Packet (see Sections 4.5.3.5 and 4.5.3.6, respectively), the Receive TEP shall transmit it immediately. The Sequence Number of the Flow Control Packet shall be the same as that of the last Packet transmitted from the Receive TEP. The Sequence Number of a Heartbeat Packet shall be set to 0.

Once the Receive TEP has transmitted a Flow Control or Heartbeat Packet, the Receive TEP shall generate and start a Transmit timer at the time when the last octet of the Packet is transmitted.

An independent Transmit timer shall be maintained for each outstanding Packet.

The initial value for the Transmit timer shall be the same as that used by the Transmit TEP of the same Transport Channel (see Section 4.4.4.5).

Upon the reception of the Ack Packet for an outstanding Packet, the Receive TEP shall discard the Transmit timer for that Packet.

If the Receive TEP has received no Ack Packet for a Packet within the period allowed by the Transmit timer for the Packet and if the retry count for the retransmission is less than or equal to the maximum retry count, the Receive TEP shall retransmit the Packet with the original Sequence Number and shall restart the Transmit timer for the Packet.

The maximum retry count shall be the same as that used by the Transmit TEP of the same Transport Channel (see Section 4.4.4.5).

受信 TEP は、本 procedure に従って、Flow Control Packets (Transport Channel に Flow Control を用いる場合) 及び/または Heartbeat Packets (Transport Channel に受信 Heartbeat を用いる場合) を送信、そして必要に応じて適宜再送信する。これらは、OPEN 状態で実行される (4.5.2.4 項参照)。

受信 TEP が、Flow Control または Heartbeat Packet を生成 (それぞれ、4.5.3.5 項及び 4.5.3.6 項参照) したら、その Packet を直ちに送信すること。Flow Control Packet の Sequence Number は、直前に受信 TEP から送信された Packet の Sequence Number と同じであること。Heartbeat Packet の Sequence Number は、0 に設定すること。

受信 TEP は、Flow Control または Heartbeat Packet を送信したら、Packet の最後の octet が送信されたタイミングで送信 timer を生成し、開始すること。

応答未確認の Packet 毎に、独立した送信 timer を維持すること。

送信 timers の初期値は、同じ Transport Channel の送信 TEP で用いる値 (4.4.4.5 項参照) と同じであること。

受信 TEP は、応答未確認の Packet に対する Ack Packet を受信したら、その Packet の送信 timer を破棄すること。

受信 TEP が、ある Packet に対する Ack Packet を受信しないままその Packet の送信 timer が時間切れになり、かつ、再送信の再試行回数が最大再試行回数以下であれば、受信 TEP はその Packet を元の Sequence Number で再送信すると共にその Packet の送信 timer を再始動すること。

最大再試行回数は、同じ Transport Channel の送信 TEP で用いる回数 (4.4.4.5 項参照) と同じであること。

If the retry count for the retransmission of a **Packet** exceeds the maximum retry count, the **Receive TEP** **shall** declare the **Transport Channel** to be “inactive”, **shall** transition into the **CLOSED** state, and, **shall** notify the receiving application, through the **ChannelControl.indication primitive** (see Section 3.2.3.3), of the fact that the **Transport Channel** has been closed.

If the **Receive TEP** receives an **Ack Packet** for a non-outstanding **Packet** (*i.e.*, a **Packet** for which no **Transmit timer** is running), the **Receive TEP** **shall** ignore it.

ある **Packet** の再送信の再試行回数が最大再試行回数を超えた場合、**受信 TEP** は、その **Transport Channel** は「非アクティブ」と宣言し、**CLOSED** 状態に移行すると共にその **Transport Channel** を閉じた事を、**ChannelControl.indication primitive** (3.2.3.3 項参照) を介して受信アプリケーションに通知すること。

**受信 TEP** は、応答確認済の **Packet** (つまり、**送信 timer** が実行されていない **Packet**) の **Ack Packet** を受信した場合、その受信を無視すること。

#### 4.5.3.8. Processing and Generation of Secondary Header // Secondary Header の処理と生成

(the same as Section 4.4.4.8)

(4.4.4.8 項に同じ)

#### 4.5.3.9. Redundancy Control Support // 冗長系制御のサポート

Once a Transmit TEP has declared the Transport Channel to be “inactive” (see Section 4.4.4.5), the sending application might choose to open an alternative Transport Channel using a separate route to the receiving node and retransmit all the outstanding Data Packets through the new Transport Channel (see Section 4.4.4.9). In the case where this type of Redundancy Control mechanism is implemented by the sending application, the Receive TEP shall use Receive Heartbeat (see Sections 4.4.4.7 and 4.5.3.6) in order to detect the failure of the communication and to close inactive Transport Channels.

ある送信 TEP が Transport Channel を「非アクティブ」であると宣言したら (4.4.4.5 項参照)、送信アプリケーションは、受信 node への別のルートを用いて代替 Transport Channel を開く事を選択し、新しい Transport Channel (4.4.4.9 項参照) を介して全ての応答未確認の Data Packets を再送信する可能性がある。この種の冗長系制御の機構が送信アプリケーションによって実装される場合、受信 TEP は、通信の障害を検出し、非アクティブな Transport Channel を閉じるために、受信 Heartbeat (4.4.4.7 項及び 4.5.3.6 を参照) を用いること。

#### 4.5.4. RECONSTRUCTION (REASSEMBLE)

The Receive TEP reconstructs transmitted service data units and delivers them to the receiving application, both according to this procedure.

One instance of this procedure shall exist for each Transport Channel.

The Receive TEP shall, according to this procedure, extract service data units (or their segments) from the Application Data fields (see Section 4.2.4.3) of accepted Data Packets (see Section 4.5.3.4), shall reconstruct the original service data units using the values of the Sequence Flags (see Section 4.2.3.4.4) of the Data Packets if necessary, and shall notify the receiving application, through the DataTransfer.indication primitive of the reconstructed service data units.

本 procedure に従って、受信 TEP は、送信された service data units を再構築し、受信アプリケーションに配信する。

本 procedure の instance は、Transport Channel 毎に一つ存在すること。

本 procedure に従って、受信 TEP は、受理した Data Packets (4.5.3.4 項参照) の Application Data フィールド (4.2.4.3 項参照) から service data units (またはその segments) を抽出し、必要に応じてそれら Data Packets の Sequence Flags (4.2.3.4.4 項参照) の値を用いて元の service data units を再構築し、再構築した service data units を、DataTransfer.indication primitive (3.3.3.3 項参照) を介して受信アプリケーションに通知すること。

## 4.6. NODE PROCEDURES

### 4.6.1. GENERAL // 一般

This section specifies the **procedures** shared by all of the **Transmit TEPs** and **Receive TEPs** that exist on a **node**.

[Note] The **procedures** specified in this subsection are presented in an abstract sense. This section does not describe any specific ways of implementation of a **protocol entity**.

The following two **procedures** **shall** be used at a **node**:

- a) Channel Multiplexing, and
- b) Channel Demultiplexing.

本項は、**node** に存在する**送信 TEPs** と**受信 TEPs** の全てが、共用する **procedures** を定める。

[注] 本項が定める **procedures** は、抽象的な意味で提示されている。本項は**プロトコル構成要素**の具体的な実装方法を記すものではない。

**Node** では、以下の二つの **procedures** を用いる**こと**。

- a) チャネル多重化
- b) チャネルの逆多重化

#### 4.6.2. CHANNEL MULTIPLEXING // チャネル多重化

A **node shall**, according to this **procedure**, multiplex the transfers of **Packets** transmitted and retransmitted by the **Transmit TEP** and **Receive TEP** of any of the **Transport Channels**. The multiplexing yields a single stream of **Packets**. The **node shall** transmit the stream on the outgoing **SpaceWire link**. Note that the **Transmit TEPs** transmit and (when required and appropriate) retransmit the **Packets** according to the **Transmission Control procedure** (see Section 4.4.4) and the **Receive TEPs** do according to the **Reception Control procedure** (see Section 4.5.3).

One **instance** of this **procedure shall** exist at each **node**.

In the stream of **Packets** mentioned above, a **Packet shall** be transmitted according to, firstly, the priority level assigned to the **Transport Channel** (if it is assigned), and secondly, the priority level assigned to the type of the **Packet**. The priority level **may** be assigned for each **Transport Channel**. The priority level **shall** be, if it is chosen to be specified. specified in a project-specific document, using the table provided in Section 5.2 of this document.

The latter priority levels of the type of the **Packets shall** be in the following order from the highest:

- 1) **Ack Packet**,
- 2) **Control Packet**,
- 3) **Data Packet**,
- 4) **Flow Control Packet**, and
- 5) **Heartbeat Packet**.

This list implies, for example, that if an **Ack Packet** is available for a transmission, the **Ack Packet** is transmitted before the **Packets** of the other types through the **Transport Channel**.

本 **procedure** に従って、**node** は、あらゆる **Transport Channels** の**送信 TEP** 及び**受信 TEP** が送信及び再送信する **Packets** の伝送を多重化すること。この多重化は、**Packets** の単一ストリームを生成する。また、**node** は、このストリームを出力 **SpaceWire link** で送信すること。なお、**送信 TEPs** は、これらの **Packets** を、**送信制御 procedure** に従って送信及び(必要に応じて適宜)再送信 (4.4.4 項参照) し、**受信 TEPs** は、これらの **Packets** を、**受信制御 procedure** に従って送信及び再送信 (4.5.3 項参照) する。

本 **procedure** の **instance** は、**node** 毎に一つ存在すること。

上記の **Packets** のストリームでは、**Packet** は、第一に **Transport Channels** に割り当てられた優先度レベル (割り当てられる場合)、第二に **Packets** の種類に割り当てられた優先度レベルに基づき送信されること。優先度レベルは、**Transport Channel** 毎に割り当てて**良い**。優先度レベルは、定めることを選択した場合、本書 5.2 項が提供する表を用いてプロジェクト固有の文書に定めること。

後者の **Packets** の種類のレベルの優先順位は、高いものから以下の順であること。

- 1) **Ack Packet**
- 2) **Control Packet**
- 3) **Data Packet**
- 4) **Flow Control Packet**
- 5) **Heartbeat Packet**

このリストは、たとえば **Ack Packet** が送信可能である場合、**Transport Channel** を介し、他の種類の **Packets** に先立ってその **Ack Packet** が送信される事を意味する。

#### 4.6.3. CHANNEL DEMULTIPLEXING // チャネル逆多重化

A **node shall**, according to this **procedure**, receive a single stream of **Packets** on the incoming **SpaceWire link** and **shall** demultiplex the stream by delivering each of the **Packets** to the **Transmit or Receive TEP** of the appropriate **Transport Channels** according to the value of the **Channel Number** contained in the **SpW-R Packet Header** (see Section 4.2.3.6). The demultiplexing yields the incoming transfer of **Packets** of all the **Transport Channels**. Note that the **Transmit TEPs** receive the **Packets** according to the **Transmission Control procedure** (see Section 4.4.4) and the **Receive TEPs** do according to the **Reception Control procedure** (see Section 4.5.3).

One **instance** of this **procedure shall** exist at each **node**.

本 **procedure** に従って、**node** は、入力 **SpaceWire link** で **Packets** の単一ストリームを受信すること。また、それらの **Packets** の各々を **SpW-R Packet Header** (4.2.3.6 項) に含まれる **Channel Number** の値に従って適切な **Transport Channels** の送信または受信 TEP に配信する事により、このストリームを逆多重化すること。この逆多重化は、全ての **Transport Channels** の **Packets** の入力伝送を生成する。なお、送信 TEPs は、これらの **Packets** を、送信制御 **procedure** に従って受信 (4.4.4 項参照) し、受信 TEPs は、これらの **Packets** を、受信制御 **procedure** に従って受信 (4.5.3 項参照) する。

本 **procedure** の **instance** は、**node** 毎に一つ存在すること。





## 5. HOW TO SPECIFY PARAMETERS FOR EACH NETWORK

### 各 NETWORK のパラメータを定める方法

#### 5.1. GENERAL // 一般

This section presents the methods for specifying the parameters of SpaceWire-R (SpW-R) used in each SpaceWire network.

The values of the parameters of SpW-R used in a SpaceWire network shall be specified by filling the table shown in the following sub-sections.

本章は、各 SpaceWire network で用いる SpaceWire-R (SpW-R) のパラメータの定める方法を説明する。

SpaceWire network で用いる SpW-R のパラメータの値は、以下の項に示される表に記入する事によって定めること。

#### 5.2. PARAMETERS USED FOR EACH SPACEWIRE NETWORK

##### 各 SPACEWIRE NETWORK に用いるパラメータ

The parameters used for each SpaceWire Network shall be specified by filling Table 5-1.

各 SpaceWire Network に用いるパラメータは、Table 5-1 に記入して定めること。

Table 5-1: Table for Specifying the Values of Parameters Used for Each SpaceWire Network  
各 SpaceWire Network に用いるパラメータを定めるための表

Option	Reference	Value	Unit or Options
Channel Numbers	4.2.3.6	Combinations of an Integer and the purpose of the Transport Channel	

#### 5.3. PARAMETERS USED FOR EACH TRANSPORT CHANNEL

##### 各 TRANSPORT CHANNEL に用いるパラメータ

The parameters used for each Transport Channel shall be specified by filling Table 5-2.

各 Transport Channel に用いるパラメータは、Table 5-2 に記入して定めること。

Table 5-2: Table for Specifying the Values of Parameters Used for Each **Transport Channel**  
 各 **Transport Channel** に用いるパラメータを定めるための表

Option	Reference	Value	Unit or Options
SLA of <b>Transmit TEP</b>	4.2		N/A
Length of <b>Destination Identifiers</b> of <b>Transmit TEP</b>	4.2		Octet
<b>Destination Identifiers</b> of <b>Transmit TEP</b>	4.2		N/A
SLA of <b>Receive TEP</b>	4.2		N/A
Length of <b>Destination Identifiers</b> of <b>Receive TEP</b>	4.2		Octet
<b>Destination Identifiers</b> of <b>Receive TEP</b>	4.2		N/A
Maximum Length of <b>Service Data Unit</b>	3.3.2.1		Octet
Maximum Length of <b>Application Data Field</b>	4.2.4.3		Octet
<b>Sliding Window Size</b> ( $k$ )	4.4.4.3		Integer
<b>Transmit Timer</b> Initial Value	4.4.4.5		Millisecond
Maximum Retry Count	4.4.4.5		Integer
<b>Flow Control</b>	4.4.4.6 and 4.5.3.5		Used or Not Used
<b>Transmit Heartbeat</b>	4.4.4.7		Used or Not Used
<b>Transmit Heartbeat Timer</b> Initial Value	4.4.4.7		Millisecond
<b>Receive Heartbeat</b>	4.5.3.6		Used or Not Used
<b>Receive Heartbeat Timer</b> Initial Value	4.5.3.6		Millisecond
<b>Close Timer</b> Initial Value	4.5.2.5		Millisecond
Priority Level	4.6.1		Integer

[Example] An example of specifying the values of the parameters of **SpW-R** used for a **Transport Channel** is shown in Appendix C.

[例] **Transport Channel** に用いる **SpW-R** のパラメータの値を定める例を付録 C に示す。

## APPENDIX A. ACRONYMS

This chapter lists acronyms used in this document.

本章では、本書が用いる略語一覧を示す。

CCSDS Consultative Committee for Space Data Systems

CRC Cyclic Redundancy Check

ECSS European Cooperation for Space Standardization

GRDDP GOES-R Reliable Data Delivery Protocol

GSFC Goddard Space Flight Center

ISO International Organization for Standardization

JAS Joint Architecture Standard

MASN Maximum Acceptable Sequence Number

NASA National Aeronautics and Space Administration

PDU Protocol Data Unit

RDDP Reliable Data Delivery Protocol

SDU Service Data Unit

SpW SpaceWire

SLA SpaceWire Logical Address

TEP Transport End Point



## APPENDIX B. EXAMPLES OF PDU EXCHANGES // PDU 交換の例

### B.1. GENERAL // 一般

This appendix shows some examples of exchanges of SpW-R Packets between the Transmit TEP and Receive TEP of a Transport Channel.

この付録は、Transport Channel の送信 TEP と受信 TEP の間の SpW-R Packets の交換の例を幾つか示す。

This appendix is not a part of the specification.

この付録は仕様の一部ではない。

### B.2. NORMAL CASE // 通常のケース

An example of exchanges of SpW-R Packets during which no errors are raised is shown in Figure B-1, where the number quoted in each pair of parentheses is the Sequence Number of the SpW-R Packet.

エラーが発生しない場合の SpW-R Packets の交換の例を Figure B-1 に示す。図中の各括弧内の番号は、当該 SpW-R Packet の Sequence Number である。

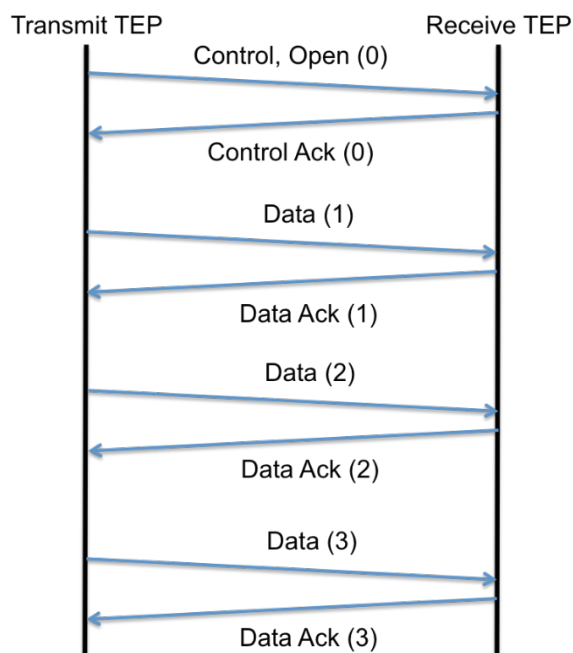


Figure B-1: Exchanges of SpW-R Packets during which no errors are raised エラーが発生しない場合の SpW-R Packets の交換

### B.3. DATA ERROR CASE // データエラーの場合

An example of exchanges of SpW-R Packets during which an error occurs in a transmission of a Data Packet is shown in Figure B-2, where the number quoted in each pair of parentheses is the Sequence Number of the SpW-R Packet.

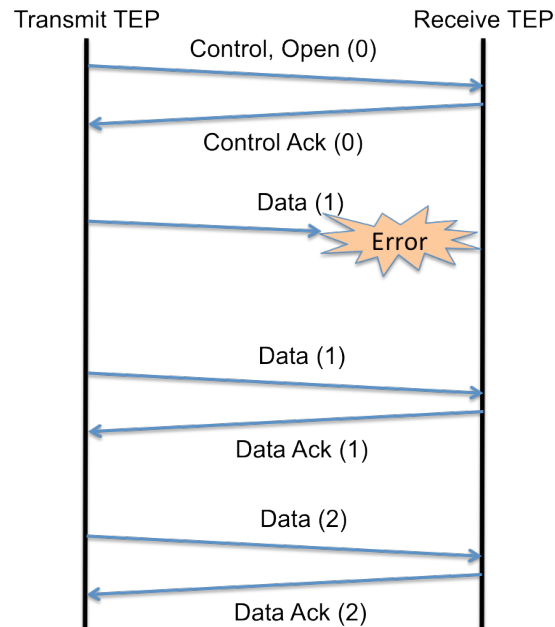


Figure B-2: Exchanges of SpW-R Packets during which an error occurs in a transmission of a Data Packet  
Data Packet の送信にエラーが発生した場合の SpW-R Packets の交換

#### B.4. ACK ERROR CASE // ACK エラーケース

An example of exchanges of SpW-R Packets during which an error occurs in a transmission of an Ack Packet is shown in Figure B-3, where the number quoted in each pair of parentheses is the Sequence Number of the SpW-R Packet.

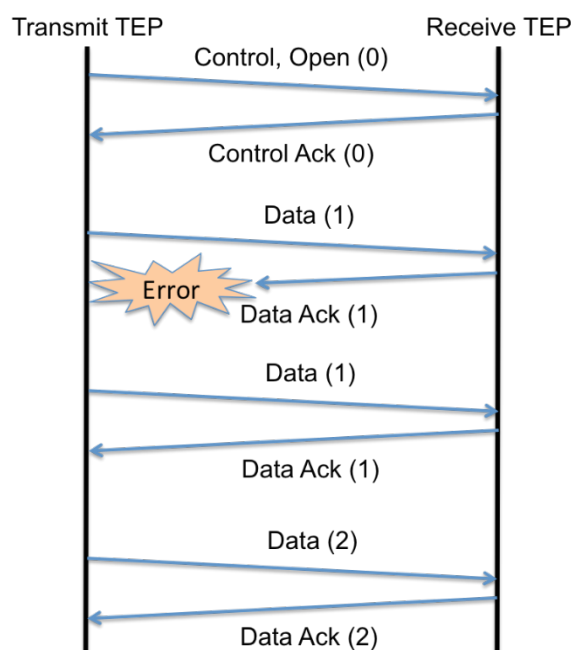


Figure B-3: Exchanges of SpW-R Packets during which an error occurs during a transmission of an Ack Packet  
Ack Packet の送信でエラーが発生した場合の SpW-R Packets の交換



## B.5. FLOW CONTROL CASE // FLOW CONTROL の場合

An example of exchanges of SpW-R Packets in which optional Flow Control is used is shown in Figure B-4. In the figure, the number quoted in a pair of the parentheses for each of the SpW-R Packets sent from the Transmit TEP to the Receive TEP (indicated with rightward-pointing arrows) is the Sequence Number of the SpW-R Packet, and the pair of the numbers quoted in a pair of the parentheses for each of the SpW-R Packets sent from the Receive TEP to the Transmit TEP (indicated with leftward-pointing arrows) are the Sequence Number and Maximum Acceptable Sequence Number (MASN), in this order, of the SpW-R Packet.

オプションの Flow Control を用いる場合の SpW-R Packets の交換の例を Figure B-4 に示す。図中、送信 TEP から受信 TEP に送る SpW-R Packets (右向き矢印で示される) においては、各括弧内に示される番号は、当該 SpW-R Packet の Sequence Number である。受信 TEP から送信 TEP に送る SpW-R Packets (左向き矢印で示される) においては、各括弧内の一組の番号は、当該 SpW-R Packet の Sequence Number と Maximum Acceptable Sequence Number (MASN) とがこの順序で示されている。

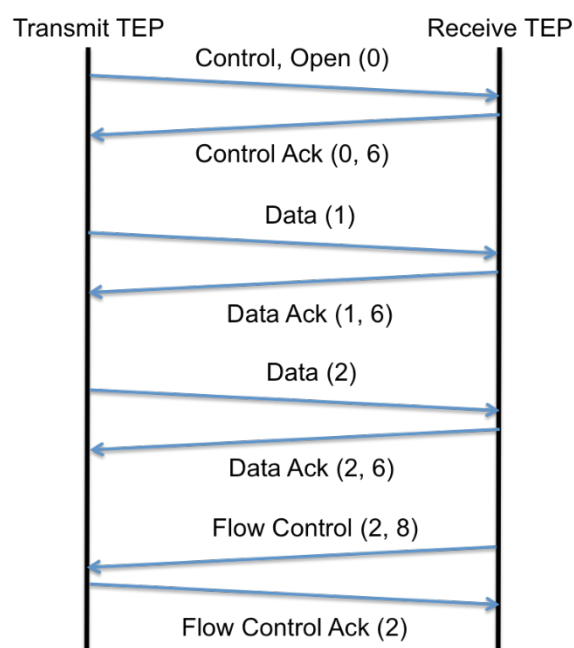


Figure B-4: Exchanges of SpW-R Packets where Flow Control is used  
Flow Control を用いる場合の SpW-R Packets の交換

## APPENDIX C. EXAMPLE OF PARAMETERS FOR A NETWORK // NETWORK のためのパラメータ例

### C.1. GENERAL // 一般

This appendix shows an example of specifying the parameters of SpaceWire-R used in a SpaceWire network. この付録は、SpaceWire network で用いる SpaceWire-R のパラメータを定める例を示す。

This appendix is not a part of the specification. この付録は仕様の一部ではない。

### C.2. PARAMETERS USED FOR A SPACEWIRE NETWORK ある SPACEWIRE NETWORK に用いるパラメータ

Table C-1: Example of Specifying the Values of Parameters Used for a SpaceWire Network

SpaceWire Netに用いるパラメータの値を定めた例

Option	Reference	Value	Unit or Options
Channel Numbers	4.2.3.6	1 (Mission Data of Camera A)	

### C. 3. PARAMETERS USED FOR A TRANSPORT CHANNEL ある TRANSPORT CHANNEL に用いるパラメータ

An example of specifying the values of the Transport Channel に用いる SpW-R のパラメータの parameters of SpW-R used for a Transport Channel 値を定める例を Table C-1 に示す。  
is shown in Table C-1.

**Table C-1: Example of Specifying the Values of Parameters Used for a Transport Channel**

Transport Channel に用いるパラメータの値を定めた例

Option	Reference	Value	Unit or Options
SLA of Transmit TEP	4.2	65	N/A
Length of Destination Identifiers of Transmit TEP	4.2	0	Octet
Destination Identifiers of Transmit TEP	4.2	N/A	N/A
SLA of Receive TEP	4.2	66	N/A
Length of Destination Identifiers of Receive TEP	4.2	0	Octet
Destination Identifiers of Receive TEP	4.2	N/A	N/A
Maximum Length of Service Data Unit	3.3.2.1	2048	Octet
Maximum Length of Application Data Field	4.2.4.3	256	Octet
Sliding Window Size ( $k$ )	4.4.4.3	8	Integer
Transmit Timer Initial Value	4.4.4.5	500	Millisecond
Maximum Retry Count	4.4.4.5	3	Integer
Flow Control	4.4.4.6 and 4.5.3.5	Used	Used or Not Used
Transmit Heartbeat	4.4.4.7	Used	Used or Not Used
Transmit Heartbeat Timer Initial Value	4.4.4.7	2000	Millisecond
Receive Heartbeat	4.5.3.6	Used	Used or Not Used
Receive Heartbeat Timer Initial Value	4.5.3.6	2000	Millisecond
Close Timer Initial Value	4.5.2.5	1600	Millisecond
Priority Level	4.6.1	2	Integer