JAXA-RPR-MX16309

RELEASE DATE: Mar.1, 2017

MARTIAN MOONS EXPLORATION (MMX) MISSION

SPACEWIRE ONBOARD SUBNETWORK DESIGN STANDARDS (DRAFT)

REVISION AND HISTORY PAGE

Status	Revision Number	Change Number	Description	Release Date

Chapter 0

Contents

Chapter 1

Abstract

1.1 Scope of application

1. This design standard is applied to areas using SpaceWire to implement CCSDS SOIS subnetwork services in spacecraft system data communication interfaces.

2. Specific methods of transmitting control commands, HK data, and observation data using a subnetwork service specified in this design standard are defined as a layer above the design standard as the SM&C layer.

1.2 Creation of these standards

The contents of this document are as follows.

- §2 presents an overview of the protocol stack stipulated by these standards.
- §3, §4, §5, §6, and §7 describe an overview of each standard, and stipulate items for consideration in each layer.
- §8 presents specific stipulations for realizing a subnetwork service using SpaceWire-RMAP.
- §9 presents specific stipulations for realizing a subnetwork service using SpaceWire-PTP.
- §10 presents specific stipulations for realizing a subnetwork service using SpaceWire-R.

1.3 Applicable documents

1. SpaceWire, ECSS-E-ST-50-12C

2. Protocol ID, ECSS-E-ST-50-51C

3. SpaceWire-RMAP, ECSS-E-ST-50-52C

4. SpaceWire-PTP, ECSS-E-ST-50-53C

5. SpaceWire-R, SCDHA 151.0.4 Issue 0.4, Japan Aerospace Exploration Agency

6. Guidelines for methods for maintaining real-time SpaceWire, NCES-SPWRT-1-101, Japan Aerospace Exploration Agency and Nagoya University

7. Spacecraft Onboard Interface Services, CCSDS 850.0-G-2, December 2013

8. Spacecraft Onboard Interface Services—Subnetwork Packet Service, CCSDS 851.0-M-1, December 2009

9. Space Packet Protocol, CCSDS 133.0-B-1, September 2003

10. Cable, ESCC Detail Specification No. 3902/003

11. Connector, ESCC Detail Specification No. 3401/029, ESCC Detail Specification No. 3401/071

1.4 Related documents

1. SpaceWire Onboard Subnetwork Design Standards, Supplementary Materials: “SpaceWire-R Protocol Performance Evaluation Standards” CPA-114021

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

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

1.5 Terms

The following are terms used in this document and their definitions:

RMAP initiator: A device or function that sends Remote Memory Access Protocol (RMAP) command packets and by read receives data from (by write sends data to) an RMAP target device. See ECSS specifications documentation for details.

RMAP target: A device or function that receives RMAP command packets, and performs read/write operations as specified in the instruction field of the packets. See ECSS specifications documentation for details.

Communication service: This term refers to a specific method for RMAP read/write procedures necessary to realize a subnetwork service. Data are transmitted between nodes by performing the various RMAP communications defined as communication services, thereby providing functions equivalent to a subnetwork service. In some cases, it is possible to use multiple communication services for realization of a single subnetwork service. Device and system designers select and implement those communication services that are actually used.

Master device: A node that controls within-network communications and itself becomes an RMAP initiator to initiate SpaceWire-RMAP communications. Satellite control computers such as the Data Handling Unit (DHU) are an example.

User device: A node that connects to a network and is controlled through communications from a master device. Mission devices controlled by the SMU or OBC are examples. When using a subnetwork service that implements SpaceWire-RMAP, user devices become RMAP targets. Depending on its features, a given user device can in some cases be viewed as both a user device and a master device. For example, mass memory can be considered as a master device when collecting data from observational equipment via SpaceWire-RMAP communications services, or as a user device when receiving control commands from the SMU.

Service data unit (SDU): Defined in CCSDS SOIS as "a unit of data passed into or out of a service interface." In these design standards, examples include CCSDS space packets, transfer frames, and virtual channel data units.

Raw data: A byte array that does not adhere to a standardized packet format. Raw data include raw measurement data values and integer time representations.

Time master device: A device that issues time information. This can be considered independent equipment, even when the same master device is used to issue control commands.

Chapter 2
Overview of the SpaceWire onboard network

2.1 Goal of definitions in these design standards

With the goal of improving interoperability and reusability through the use of SpaceWire between onboard equipment, these design standards stipulate methods for using SpaceWire and higher protocols to realize portions equivalent to the subnetwork service of the subnetwork layer in the CCSDS SOIS layer (Fig. 2.1).

In terms of packet services, this document stipulates functions for transmitting service data units via SpaceWire in response to requests from layers above CCSDS SOIS.

In terms of memory access services, features for accessing device memory in response to requests from layers above CCSDS SOIS are provided.

In terms of synchronization services, methods for transmitting time information from time master devices to time user devices in the SpaceWire network are stipulated, and synchronization services TIME.request and TIME.indication can be implemented on the time user device side.

2.2 Protocol stack

Multiple upper protocols are defined for SpaceWire, and can be used according to the intended application. This document stipulates methods for realizing subnetwork services by using the following upper communications protocols:

- SpaceWire-RMAP,
- SpaceWire-PTP, and
- SpaceWire-R.

As needed, time division communication control using TimeCode can be used for SpaceWire physical layers and SpaceWire network quality of service (QoS).

Among the above communication protocols, only SpaceWire-R implements segmentation blocking of data within the protocol itself. When segmentation blocking is required in other protocols, it must be defined in upper protocols.

Figure 2.2 shows the protocol stack used in these standards.

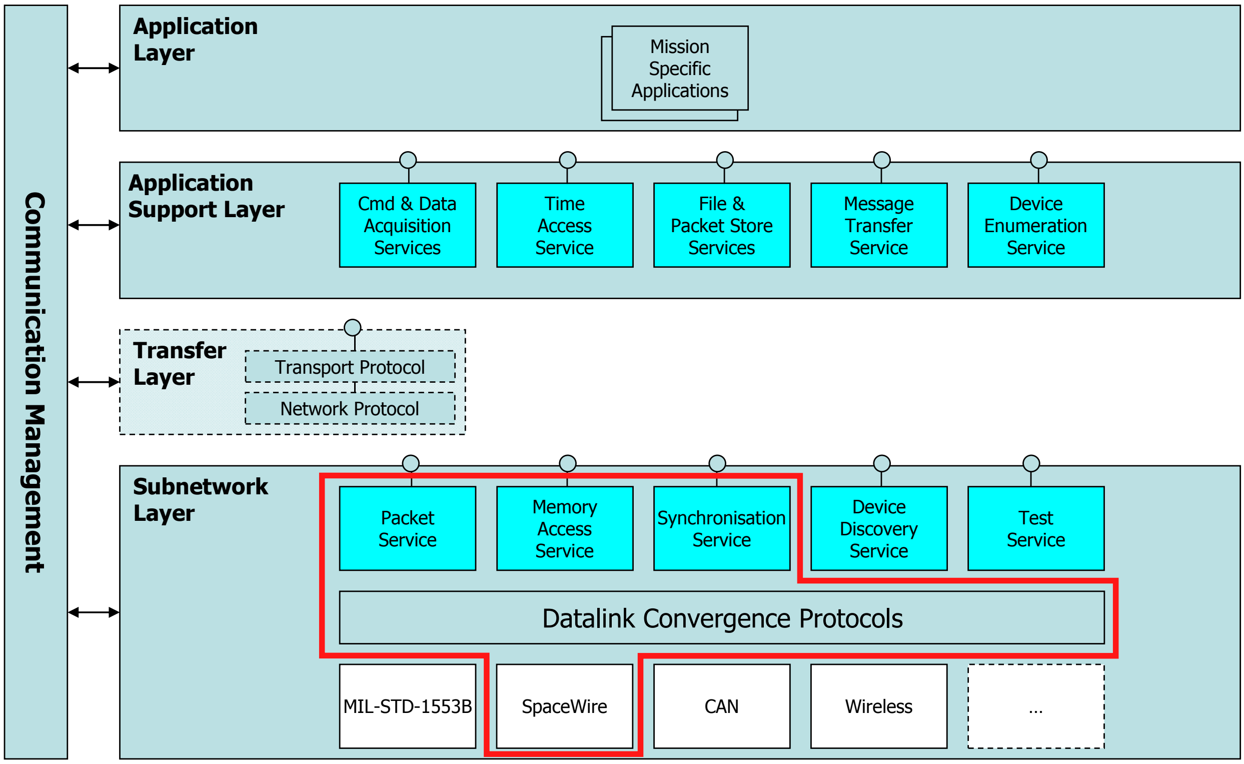

Fig 2.1: CCSDS SOIS reference architecture. Red lines indicate areas covered in these standards (From Applicable Document 7).

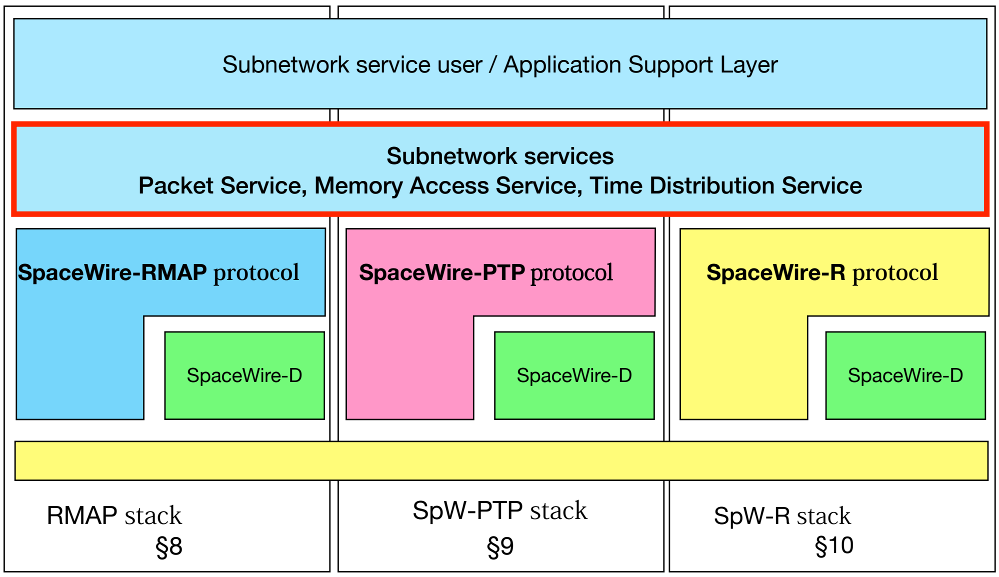

Fig 2.2: Overview of the protocol stack defined in the SpaceWire design standards

Chapter 3

SpaceWire

SpaceWire is defined by ECSS-E-ST-50-12C (replaces ECSS-E50-12A) (Applicable Document 1). SpaceWire layers require definitions according to those standards. When the standards in this document are expanded, or when applied with limitations, only those areas that could lead to misunderstandings are described.

3.1 Physical layer

3.1.1 Cables

SpaceWire cables are defined in ESCC 3902/003 (Applicable Document 10). Cables defined in ESCC 3902/003 may be difficult to use under some mission requirements. In such cases, use of electrically equivalent cables is permitted.

3.1.2 Cable assemblies

Cables specified in ESCC 3902/003 have inner and outer shields. In ECSS-E-ST-50-12C (Applicable Document 1), the outer shield connects to only the frame ground and the inner shield to only the LVDS driver ground through pin 3 on the LVDS driver, with the LVDS receiver-side left open (type AL). Also possible is a cable assembly that grounds the inner shield to the frame ground (type A). In this case, nothing attaches to pin 3.

In type AL, inner shields with differing signal directionality are electrically independent. Therefore, connectors for extending cables must provide segregated relays. Extensions thus must have connectors with more than nine pins. See Fig. 3.1 for an example.

3.1.3 Connectors

Connectors are defined in ESCC 3401/029 and ESCC 3401/071 (Applicable Document 11). Connectors and boards are connected via twisted leads. It is possible to use nonstandard connectors, but compatibility must be separately demonstrated. Their use within the housing should be limited, such as when it is necessary to use conversion cables to connect test equipment.

3.1.4 Signal levels

ECSS-E-ST-50-12C (Applicable Document 1) stipulates the utilization of LVDS for electric signal levels in SpaceWire. The LVDS common mode voltage between devices connected via SpaceWire must be within the common mode tolerance of the LVDS receiver used in the device. Items related to LVDS and electronic circuit design must follow the applicable electric design standards.

On circuit boards and within housings, signals other than LVDS can be used for SpaceWire signal levels. When this is done, signals must be within the ranges stipulated in "6.6.4 Effects of skew and jitter."

3.1.5 Link speed

In SpaceWire, other packets will not enter while a packet is being sent, so circuits are held from packet send initiation until an end-of-packet (EOP) is sent. Retention must therefore be minimized to prevent send delays. Bit rates can be freely specified in SpaceWire, and so will vary in circuits. Circuits in a slow link are retained from the data send initiation until completion, so slow links can become bottlenecks to faster links. It is thus preferable that packet circuits use links with the same bit rate.

The SpaceWire standards allow sending NULL packets during data transfer, so data may be sent intermittently. It is therefore necessary to define not only bitrates for data transfer, but also delivery times.

Bitrates in SpaceWire can be changed after link establishment. Lengths (duty ratios) can also vary by bit. Decoders for SpaceWire signals therefore cannot be designed under the assumption of a fixed bit rate.

3.1.6 Flow Control

SpaceWire uses FCT to control link flow. The timing for sending FCT varies with implementation. Implementations must accommodate the simultaneous sending of multiple FCTs.

Data transmission will stop if FCT is lost. For problem diagnosis, it is therefore preferable to monitor FCT state, such as the number of FCTs being sent and received.

3.2 TimeCode

TimeCode is issued across the entire system, allowing synchronization to within microsecond precision. Using TimeCode thus allows for synchronizations of operations across the system. TimeCode delay and jitter will depend on the number of routers passed through. The amount of added jitter will be equivalent to the time required for one character to pass through a single router. It is preferable that calculations and measurements be performed in advance.

TimeCode can represent only six bits of time information, so times requiring the use of more bits must be sent using normal SpaceWire packets, such as SpaceWire-RMAP writes from the time-issuing device to the receiving device.

3.2.1 TimeCode loss

When component-internal clocks and timers are synchronized to TimeCode received via a SpaceWire interface, operations for short- and long-term TimeCode loss and restoration must be defined at the system level.

3.3 Notes regarding SpaceWire

3.3.1 Notes regarding SpaceWire nodes

3.3.1.1 Buffering

Flow control in SpaceWire requires information exchange through links, which in turn requires buffers that consider response times. For example, efficient FCT flow control requires the use several FCTs, rather than just one. To that end, a buffer of 32-64 characters will be required, depending on the number of FCTs in simultaneous use. Improving performance will require the buffer to be located on the same chip as the CODEC. In CPU-based packet reception, a hardware buffer of several hundred to several thousand kilobytes may be needed, depending on CPU interrupt times. Several of these will be needed, so that data can be received during CPU response times. SpaceWire places no limits on packet sizes, so large packets can require the use of multiple buffers to receive data. Achieving high performance will require data transfer in which the CPU need not intervene.

3.3.2 Notes regarding SpaceWire routers

3.3.2.1 Watchdog timer

When a SpaceWire device encounters an error during packet transfer, the transfer cannot be completed and the packet retains the routing circuit. Ending this state requires the implementation of a watchdog timer on the routing switch. When a timeout occurs, an EEP error packet is sent in place of an EOP. The time until timeout is system dependent. Timeouts should preferably be counted for each port. It is also necessary to allow disabling the timeout in appropriate cases.

3.4 Stipulations related to network design

3.4.1 Network topology

Point-to-point, star, tree, and daisy chain (ring) topologies are possible. Small systems in which a single router is connected to each node (a star topology) are equivalent to a point-to-point network, so long as no node contention occurs. Operations under a single master are point-to-point equivalent, regardless of network topology. In the multi-master case, traffic from each master should use a mutually isolated topology to the extent possible.

Redundancy can be simply implemented in star or tree topologies. Robustness against one-link failures is naturally realized in ring topologies.

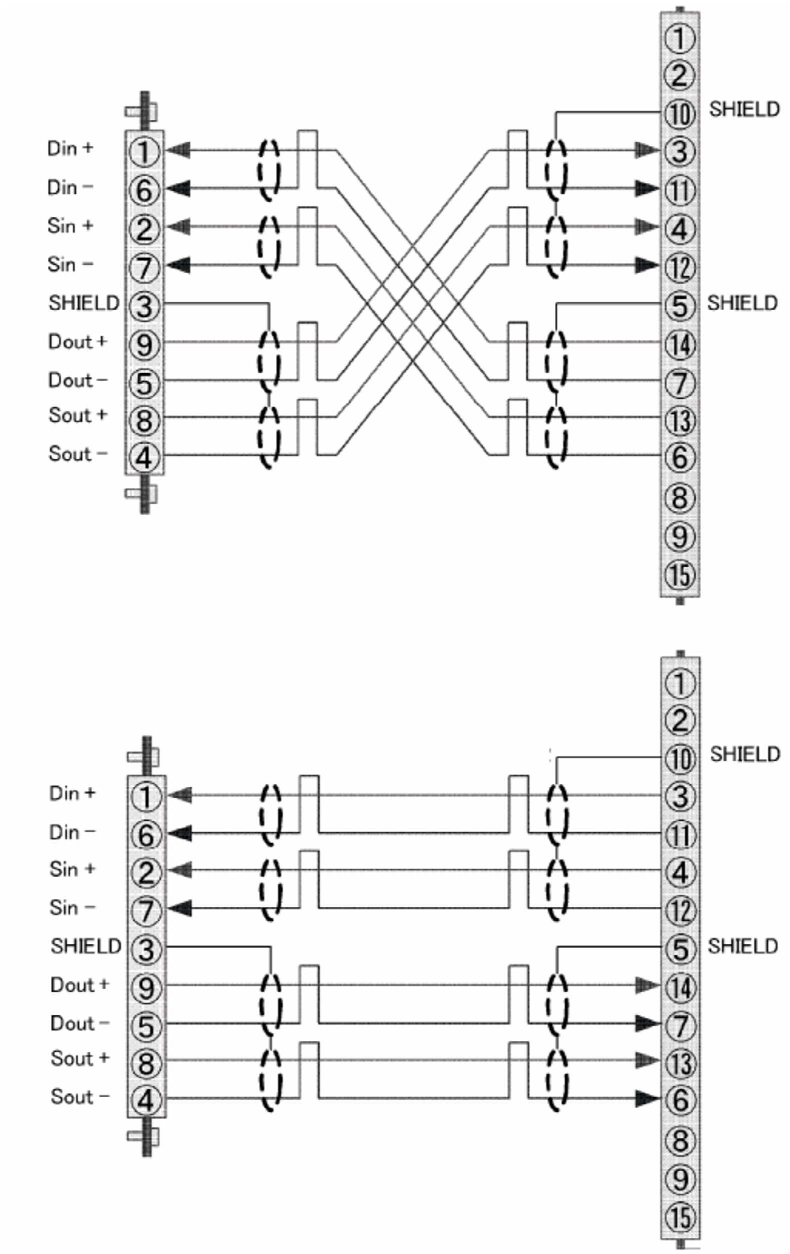

Fig 3.1: SpaceWire relay connector

Chapter 4

Time division control using TimeCode

Network time division sharing using TimeCode provides a representative design method for guaranteeing real-time, deterministic packet transfer in a SpaceWire network.

As of 2015, ESA and the SpaceWire Working Group are discussing standards for SpaceWire-D (“deterministic”), and expect ECSS documentation within a few years.

JAXA scientific satellite projects such as SPRINT-A, Hayabusa 2, and ASTRO-H have used a time division communication method established by JAXA and NEC. The approach and design methodology in this method were refined by JAXA and Nagoya University, and published as the “Guidelines for Retaining Real Time Properties in SpaceWire” (Applicable Document 6).

When adopting SpaceWire networks in spacecraft projects requiring real-time performance, projects should consider the SpaceWire-D standards and the time division communication method defined in the “Guidelines for Retaining Real Time Properties in SpaceWire” as pertaining to project-specific real-time performance requirements.

Chapter 5

SpaceWire-RMAP

This chapter presents an overview of the SpaceWire-RMAP protocol, along with related stipulations and notes. This information is presented separately because it provides fundamental stipulations for the realization of subnetwork services using SpaceWire-RMAP as described in §8.

5.1 Protocol overview

SpaceWire-RMAP is a transaction-type communications protocol that exchanges data by command-response. It adopts the following packet format and communication transactions.

5.1.1 Packet format

Figures 5.1-5.4 show the packet format used in RMAP.

5.1.2 Communication transactions

Figures 5.5 and 5.6 show typical transactions for RMAP reads and writes.

5.1.3 Error status

When an error occurs during an RMAP transaction, an error number is returned as the reply status (Fig. 5.7).

As an example, Fig. 5.8 shows the case of a RMAP write transaction in which an error reply is returned to the RMAP initiator.

First byte transmitted

	Target SpW Address	....	Target SpW Address
Target Logical Address	Protocol Identifier	Instruction	Key
Reply Address	Reply Address	Reply Address	Reply Address
Reply Address	Reply Address	Reply Address	Reply Address
Reply Address	Reply Address	Reply Address	Reply Address
Initiator Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Address
Address (MS)	Address	Address	Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	...	...	Data
Data	Data CRC	EOP	

Last byte transmitted

Bits in Instruction Field

MSB						LSB
Reserved = 0	Command = 1	Write = 1	Verify data(1) Don't Verify (0)	Reply (1)/ No reply (0)	Increment (1)/ No inc (0)	Reply Address Length
Packet Type		Command				Reply Address Length

Fig 5.1: RMAP write command packet format (From Applicable Document 3).

First byte transmitted

	Reply SpW Address	....	Reply SpW Address
Initiator Logical Address	Protocol Identifier	Instruction	Status
Target Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Header CRC
EOP			

Last byte transmitted

Bits in Instruction Field

MSB						LSB
Reserved = 0	Reply = 0	Write = 1	Verify data (1) Don't Verify (0)	Reply = 1	Increment (1)/ No inc (0)	Reply Address Length
Packet Type		Command				Reply Address Length

Fig 5.2: RMAP write reply packet format (From Applicable Document 3).

First byte transmitted

	Target SpW Address	....	Target SpW Address
Target Logical Address	Protocol Identifier	Instruction	Key
Reply Address	Reply Address	Reply Address	Reply Address
Reply Address	Reply Address	Reply Address	Reply Address
Reply Address	Reply Address	Reply Address	Reply Address
Initiator Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Address
Address (MS)	Address	Address	Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
EOP			

Last byte transmitted

Bits in Instruction Field

MSB ... LSB

Reserved = 0	Command = 1	Read = 0	Verify data = 0	Reply = 1	Increment (1) / No inc (0)	Reply Address Length
Packet Type		Command				Reply Address Length

Fig 5.3: RMAP read command packet format (From Applicable Document 3).

First byte transmitted

	Reply SpW Address	....	Reply SpW Address
Initiator Logical Address	Protocol Identifier	Instruction	Status
Target Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Reserved = 0
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	....	....	Data
Data	Data CRC	EOP	

Last byte transmitted

Bits in Instruction Field

MSB ... LSB

Reserved = 0	Reply= 0	Read = 0	Verify Data = 0	Reply = 1	Increment (1) / No inc (0)	Reply Address Length
Packet Type		Command				Reply Address Length

Fig 5.4: RMAP read reply packet format (From Applicable Document 3).

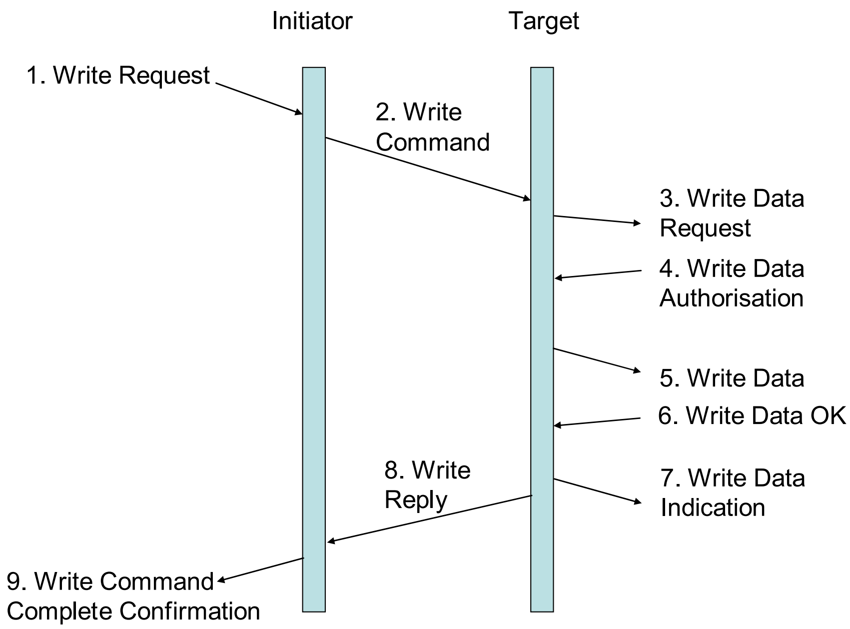

Fig 5.5: RMAP write transaction (From Applicable Document 3).

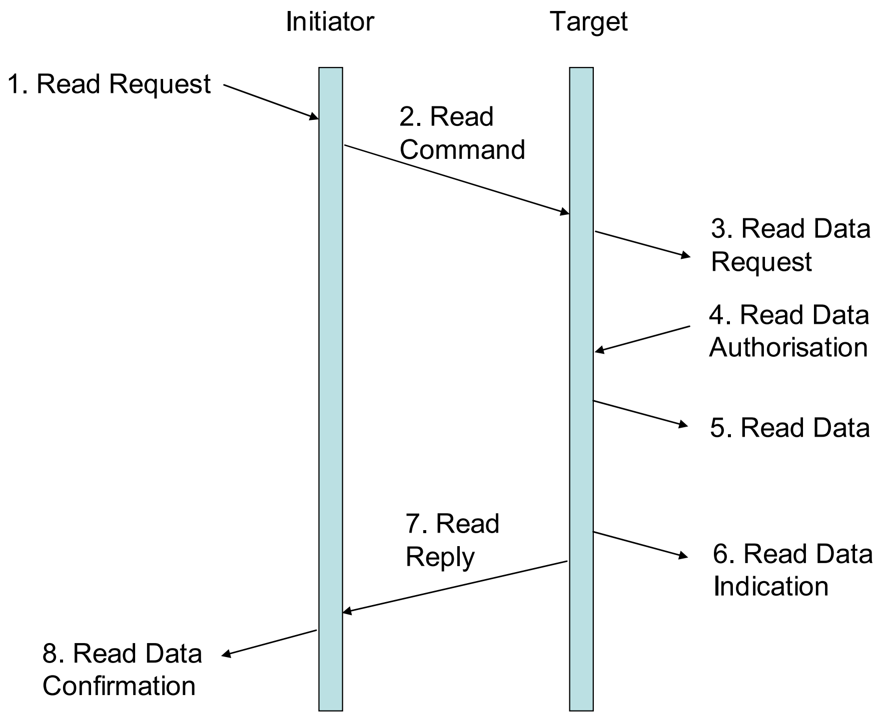

Fig 5.6: RMAP read transaction (From Applicable Document 3).

Error code	Error	Error description	Applicability		
			Write	Read	RMW
0	Command executed successfully		X	X	X
1	General error code	The detected error does not fit into the other error cases or the node does not support further distinction between the errors	X	X	X
2	Unused RMAP Packet Type or Command Code	The Header CRC was decoded correctly but the packet type is reserved or the command is not used by the RMAP protocol.	X	X	X
3	Invalid key	The Header CRC was decoded correctly but the device key did not match that expected by the target user application	X	X	X
4	Invalid Data CRC	Error in the CRC of the data field	X		X
5	Early EOP	EOP marker detected before the end of the data	X	X	X
6	Too much data	More than the expected amount of data in a command has been received	X	X	X
7	EEP	EEP marker detected immediately after the header CRC or during the transfer of data and Data CRC or immediately thereafter. Indicates that there was a communication failure of some sort on the network	X	X	X
8	Reserved	Reserved			
9	Verify buffer overrun	The verify before write bit of the command was set so that the data field was buffered in order to verify the Data CRC before transferring the data to target memory. The data field was longer than able to fit inside the verify buffer resulting in a buffer overrun Note that the command is not executed in this case	X		X
10	RMAP Command not implemented or not authorised	The target user application did not authorise the requested operation. This may be because the command requested has not been implemented	X	X	X
11	RMW Data Length error	The amount of data in a RMW command is invalid (0x01, 0x03, 0x05, 0x07 or greater than 0x08)			X
12	Invalid Target Logical Address	The Header CRC was decoded correctly but the Target Logical Address was not the value expected by the target	X	X	X
13-255	Reserved	All unused error codes are reserved			

Fig 5.7: RMAP status values defined in the protocol specification (From Applicable Document 3).

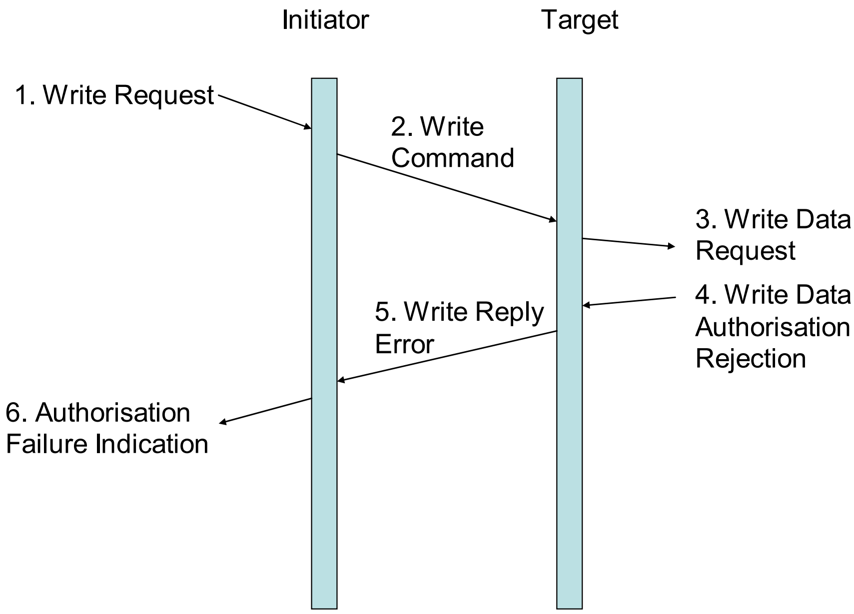

Fig 5.8: Example RMAP write transaction in which an error has occurred (From Applicable Document 3).

5.2 Overall stipulations

Applicable Document 3 should be applied when implementing SpaceWire-RMAP based on these design standards. Of the optional standards, the following should be implemented:

1. extended addressing is not used, and fixed as 0x00; and

2. read-modify-write is not used.

The following sections present notes for consideration during design.

5.3 Notes regarding initiators

1. When a given initiator is executed in parallel with multiple RMAP transactions, transactions should be separately identified by transaction ID keys.

2. In cases where an invalid transaction ID (one other than the ID for the transaction currently being managed as in progress) is received, reply packet processing (copying read reply data to memory, etc.) should preferably not be performed; instead, the upper-level application should be notified with an error.

5.4 Notes regarding targets

The following should be implemented for design simplification and improved efficiency of verification operations.

1. Keys should not be changed by memory region for a single RMAP target function.

2. All error statuses should be implemented.

3. The word width for accessing RMAP memory regions should, as a rule, be 32 bits.

   (a) When devices use word widths other than 32 bits, detailed documentation should be prepared at the system design stage.

   (b) To prevent any discrepancies, documentation should include text dumps of RMAP command and reply packets for the case of accessing the relevant device using word widths other than 32 bits.

4. Data byte alignment when accessing via RMAP should be big-endian. Note that this means that data should be big-endian when constructing RMAP packets; it does not specify the byte alignment of data in CPU memory or in FPGA memory or registers.

5.5 Instructions

1. Implemented instructions other than read-modify-write should be presented in ICD and other documentation as a combination of the following.

   - Non-verify write (with reply)
   - Non-verify write (without reply)

- Verify write
- Read

2. For RMAP-accessible memory spaces, the set of instructions allowed access should be presented in ICD documentation.

5.6 Addressing

1. Byte (not word) addressing should be used for memory spaces accessed by RMAP.

   Ex.: In the case of 32 bits per word, where the first word starts at 0x0000_0000, the second word starts at 0x0000_0004.

Chapter 6

SpaceWire-PTP

This chapter describes an overview of the SpaceWire-PTP protocol, with related stipulations and notes. This information is presented separately because it provides fundamental stipulations for the realization of subnetwork services using SpaceWire-PTP as described in §9.

6.1 Protocol overview

SpaceWire-PTP as stipulated in Applicable Document 4 is a protocol for unidirectional sending of CCSDS space packets between SpaceWire-connected nodes. CCSDS space packets are stored in the cargo portion of SpaceWire packets, and delivered to target nodes as SpaceWire packets.

SpaceWire-PTP packet transmission does not provide QoS such as acknowledgments, retransmission control, or latency guarantee. When these are needed, they must be provided from upper layers (or, in some cases, by use of the SpaceWire-R protocol).

6.1.1 Packet format

Figure 6.1 shows the SpaceWire-PTP packet format.

1. The SpaceWire-PTP protocol ID is 0x02 (see Applicable Document 2).
2. CCSDS space packets sent by SpaceWire-PTP must have size between 7 and 65,542 bytes (octets).
3. Status code values and meanings are as follows:

   0x00 Normal CCSDS space packet within a SpaceWire-PTP packet
   0x01 SpaceWire-PTP packet received at EEP termination
   0x02 The reserved field in the SpaceWire-PTP packet is nonzero

6.1.2 SpaceWire-PTP operations

Figure 6.2 shows SpaceWire-PTP packet send and receive operations.

6.2 Overall stipulations

1. Applicable Document 4 should be applied when implementing SpaceWire-PTP following these design standards.

First byte transmitted

	Target SpW Address	.	Target SpW Address
Target Logical Address	Protocol Identifier	Reserved = 0x00	User Application
CCSDS Packet (First Byte)	CCSDS Packet	CCSDS Packet	CCSDS Packet
CCSDS Packet	...	...	CCSDS Packet
CCSDS Packet	CCSDS Packet (Last Byte)	EOP	

Last byte transmitted

Fig 6.1:　SpaceWire-PTP packet format (from Applicable Document 4).

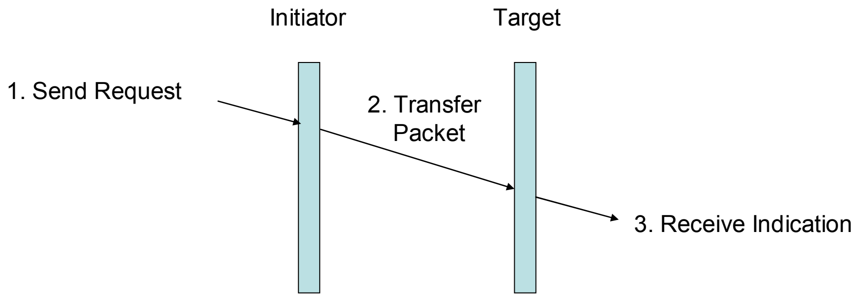

Fig 6.2:　SpaceWire-PTP send and receive operations (from Applicable Document 4).

2. The control parameters listed in Table A-1 of Applicable Document 4 should be presented in ICD documentation.

Chapter 7
SpaceWire-R

This chapter gives an overview of the SpaceWire-R protocol, with related stipulations and notes. This information is presented separately because it provides fundamental stipulations for the realization of subnetwork services using SpaceWire-R as described in §10.

7.1 SpaceWire-R overview

SpaceWire-R as defined in Applicable Document 5 is a protocol for reliable, high-speed data transmission between applications onboard satellites. Reliability in this case refers to assurance that data from the sending application arrives intact at the receiving application, and includes confirmation to the sender that the receiver is alive, and vice versa.

SpaceWire-R adds several features to the Joint Architecture Standard Reliable Data Delivery Protocol (JAS RDDP; Related Document 2) developed by the Sandia National Laboratory, USA. Differences between SpaceWire-R and JAS RDDP are described in detail in Applicable Document 5, and the two protocols are compatible (can be used to connect to each other) if the features newly added to SpaceWire-R are not used. The JAS RDDP is an extension of the GOES-R Reliable Data Delivery Protocol (GRDDP; Related Document 3), developed by NASA's Goddard Space Center. However, JAS RDDP and GRDDP use different header formats, and so are mutually incompatible and prevent interconnection unless format conversions are applied. For the same reason, SpaceWire-R and GRDDP are incompatible unless format conversions are applied.

Section 7.2 describes the overall protocol structure when using SpaceWire-R. Section 7.3 presents typical SpaceWire-R usage examples. SpaceWire-R provides several features besides reliability, and an overview of each is given in overview in §7.4. SpaceWire-R provides these features as standard services, and §7.5 describes an overview of these services. Section 7.6 presents the packet types and their formats used in the SpaceWire-R protocol, and §7.7 provides cautions when using SpaceWire-R. Section 7.8 summarizes overall stipulations regarding the use of SpaceWire-R.

7.2 Protocol configuration when using SpaceWire-R

SpaceWire-R establishes a virtual transmission path (called a transport channel) between implementing applications, and allows sending applications to sequentially transmit data chunks (typically, CCSDS space packets, etc.) to receiving applications.

Figure 7.1 shows the overall basic protocol configuration when using SpaceWire-R.

As Fig. 7.1 shows, SpaceWire-R is used as an upper-level protocol for SpaceWire (see chapter 3). In other words, data units generated by SpaceWire-R (called SpaceWire-R packets) are transmitted as cargo in

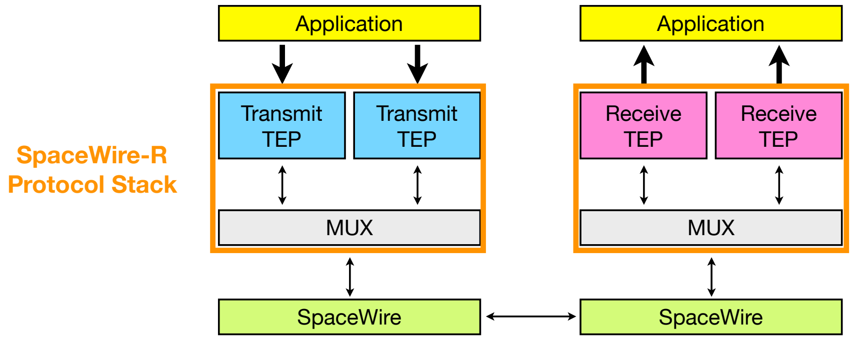

Fig 7.1: Block diagram of the SpaceWire-R protocol stack, with an example of two transmit and receive TEPs implemented on the left and right nodes.

SpaceWire packets, to which all SpaceWire stipulations are applied. Figure 7.2 shows the relation between SpaceWire-R and SpaceWire packets.

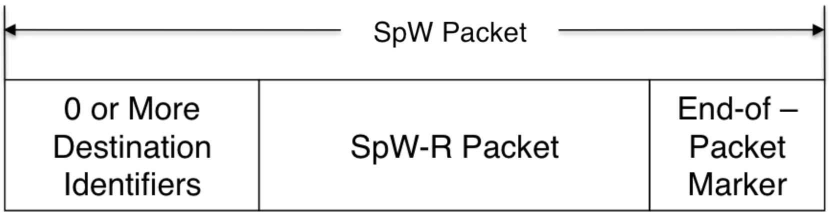

Fig 7.2: Relations between SpaceWire-R and SpaceWire packets.

SpaceWire-R presumes that data generated by the sending application are directly received by the receiving application, so there is no need for the application to write data to specific memory locations during transmission. In other words, when using SpaceWire-R, there is no need to use RMAP (see Chapter 5).

As described above, the stipulations of SpaceWire are applied when using SpaceWire-R, so it is also possible to perform one-to-many or many-to-many transmission by using a router and SpaceWire addresses on a single SpaceWire network. In this case, a different transport channel is set for each transmission-reception pair. Multiple transport channels may also be set for one transmission-reception pair.

When using a router, SpaceWire-D (a protocol for time division multiplexing using time slots, being considered by the SpaceWire International Committee) can be used to avoid router congestion. In that case one or more transport channels will be assigned to one or more SpaceWire-D time slots.

7.3 Application of SpaceWire-R

The following are some typical examples of applying SpaceWire-R.

1. Transfer of raw observation data from onboard observation equipment to processors for observation data processing, such as when transferring images captured by on-board cameras to a processor for image processing. In such cases, image data are directly transmitted via SpaceWire-R without using CCSDS space packets, etc.

2. Transfer of processed observation data from processing computers to large-capacity storage devices, such as to data recorders for storage before transfer to the ground. In this case, processed image data are transmitted via SpaceWire-R as sequences of CCSDS space packets.

3. Transfer of downlink data from large-capacity storage devices to communication devices, such as when forwarding data stored in data recorders to communication devices for transfer to the ground. In this case, too, data stored in large-capacity storage devices are sent via SpaceWire-R as CCSDS space packets.

7.4 SpaceWire-R features

The following is an overview of SpaceWire-R features.

7.4.1 Multiplexing

SpaceWire-R can collect (multiplex) and transmit multiple separate data streams sent from a given SpaceWire logical address to another SpaceWire logical address. Each data stream is assigned to one transport channel. Each transport channel is controlled independently, and data transmission characteristics can be set for each transport channel. It is also possible to set priorities for each transport channel. Section 7.6 describes specific examples of multiplexing by transport channels.

7.4.2 Data division

If the sending application (Fig. 7.1) provides SpaceWire-R with a chunk of data larger than the data unit (a SpaceWire-R packet), SpaceWire-R can divide that data chunk into multiple chunks that can be transmitted by SpaceWire-R packets. When the receiving side receives the divided data, the original data chunk is restored and delivered to the receiving-side application.

7.4.3 Reliability

SpaceWire-R guarantees that sending applications (Fig. 7.1) provide receiving applications the data with no omissions, no duplication, and in the order sent. To realize this, SpaceWire-R confirms that sent packets were properly received, and resends any packets that were not.

7.4.4 Flow control (optional)

As an optional feature, receivers can notify senders of how many more SpaceWire-R packets they can accept. This feature allows receivers to avoid receiving more SpaceWire-R packets than they can handle.

7.4.5 Heartbeats (optional)

As an optional feature, SpaceWire-R can allow confirmation that senders or receivers still have a live connection to the other, even when there are no data to be sent or received.

7.5 SpaceWire-R services

The following describes an overview of services that SpaceWire-R can provide to applications. Here, "services" refers to features provided to applications via SpaceWire-R's application program interface (API).

- Channel control service: A service for controlling the state of transmit and receive TEP
- Data transfer service: A service for sending and receiving data via TEP

7.5.1 Channel control service

This service allows applications to control the state of the SpaceWire-R protocol. Specifically, the application can use this service to instruct the SpaceWire-R protocol to open or close a particular transport channel (see Applicable Document 5). There is also a feature that allows notifying applications of the current state from the SpaceWire-R protocol.

The following is an overview of primitives defined in the channel control service.

- ChannelControl.request: Used to pass requests to TEPs so that upper-level applications can change the state of transmit or receive TEP used in a given transport channel.
- ChannelControl.indication: Used to notify upper-level applications that the state of transmit or receive TEPs has changed.

7.5.2 Data transfer service

This service allows applications to use the SpaceWire-R protocol to transfer data.

The sending-side application hands data chunks to be transmitted using this service to SpaceWire-R. SpaceWire-R notifies the application of the transfer result of each data chunk (to senders, whether the data were properly accepted, and, to receivers, whether the data were properly received).

The receiving application uses this service to accept received data chunks via SpaceWire-R.

The following is an overview of primitives defined in the data transfer service.

- DataTransfer.request: Used by the transmitting application to issue a data transmission request to the transmit TEP
- DataTransferNotify.indication: Used by the transmit TEP to notify data-sending applications of events related to SDU transmission events
- DataTransfer.indication: Used to notify data-receiving applications of SDUs received from the receive TEP

7.6 Packet type and format

The SpaceWire-R protocol uses the packets listed in Table 7.1 to realize data transmission. Individual SpaceWire-R packets are represented by storing information in the cargo part of SpaceWire packets in the format shown in Fig. 7.3.

Table 7.1: SpaceWire-R packet types.

Packet Type	Description
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 and Flow control Ack packet
7	Control Ack packet

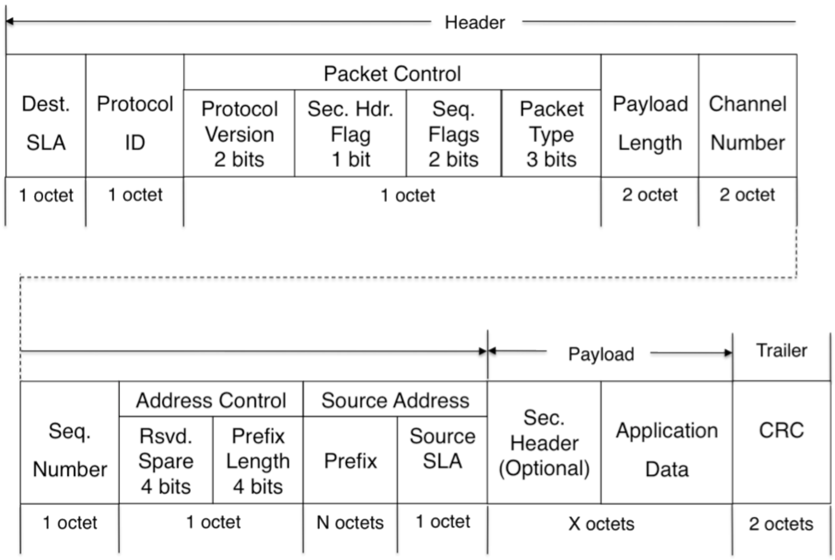

Fig 7.3: SpaceWire-R packet formats (From Applicable Document 5).

7.7 Points of note when using SpaceWire-R

The following is a simple description of transport channels.

As an example, data forwarded to satellites use commands and housekeeping data for real-time satellite control, and while some data are strongly affected by transmission delays, data such as observation or memory data are not, so long as they are forwarded correctly. In this way, different data types demand different kinds of QoS, and transport channels can be used to realize such different service qualities.

In the case of the above examples, transport channels for real-time control and those for observation results and memory data are fixed, with the former having a higher priority than the latter. When both channels need to send data, the higher-priority channel sents first. Control parameters can also be independently set by transport channel, with the higher-priority channel having a lower initial timer value (see Applicable Document 5 for details). Doing so means that should a transmit error occur in the higher-priority transport channel, data will be resent at shorter intervals, allowing control with a suitable level of required service.

7.8 Overall stipulations

1. Implementations of SpaceWire-R following these design standards must implement Applicable Document 5.
2. Optional selections among those summarized in Chapter 5 of Applicable Document 5 must be documented in the ICD.

Note See Related Document 1 regarding performance evaluation results (communication speeds, latency, etc.) when adopting the SpaceWire-R protocol.

Chapter 8

SpaceWire-RMAP subnetwork services

8.1 Overview

The following is an overview of SpaceWire-RMAP subnetwork services.

1. Because SpaceWire-RMAP subnetwork services use RMAP Readand RMAP Writeto transmit data to upper layers, the following CCSDS SOIS (Applicable Document 7) subnetwork layer services can be provided:

   (a) Packet services

   (b) Memory access services

   (c) Time information distribution required by synchronization services

2. When real-time transmission of RMAP packets must be guaranteed, time-division network sharing using SpaceWire-D is adopted as a lower layer.

3. Subnetwork services using RMAP are assumed to be mainly used for satellite bus systems (device control, housekeeping data collection, etc.) because of their compatibility with real-time property assurance and ease of implementation in user devices. Depending on the results of system design, if the request (communication speed, etc.) is satisfied by a subnetwork service using RMAP, it may be used for transmission of observation data from the mission sensor.

4. Since RMAP itself has no retransmission control mechanism, retransmission control using RMAP replies is used to realize the assured (having retransmission control) and guaranteed (having a real-time nature and retransmission control) resource reservation functions of subnetwork services as stipulated by CCSDS SOIS. (Since retransmission control can be performed in the upper SM&C layer without using retransmission control in the RMAP layer, use or non-use of retransmission in the RMAP layer is a system design option.)

8.1.1 Overall stipulations

Overall provisions of subnetwork services by SpaceWire-RMAP are presented below.

1. To reduce variations in system design and improve interoperability, communication services are realized only from the master device to the user device by pull-type data communication. That is, the master device is always the RMAP initiator, and the user device always operates as the RMAP target.

2. Polling from a master device is used for user devices to notify master devices of data transmission requests. See "User Trigger Transmission Services" for details.

3. The format of transmitted data must be service data units or raw data. See Table 8.2 for correspondences between communication services and data formats.

4. Select and determine services for transmission of various data (telemetry or command) for each user device, together with the values of relevant parameters.

5. The use or nonuse of real-time property guarantees by SpaceWire-D should be decided at the system design stage.

6. The system design determines the presence or absence of retransmission control for each communication type (for example, writing TC packets, reading HK data in TM packet format, reading of TM packets including sensor telemetry, etc.) required in the upper SM&C layer.

7. For each communication service, allocate a different memory area in the RMAP memory space and use it for data transfer.

8.1.2 Supplied subnetwork services

The following subsections describe an overview of the following subnetwork services:

1. Packet services

2. Memory access services

3. Time information distribution required by synchronization services

The details of each subnetwork service are stipulated in §8.2, §8.3, and §8.4, respectively.

8.1.2.1 Packet services

1. Provides features for transmitting service data units between nodes.

2. The following communication services are specified according to the communication method:

    - Master-trigger data transfer
        - Master-trigger SpacePacket write service
          (Best effort with no retransmission control, assured with retransmission control)
        - Master-trigger SpacePacket read service
          (Best effort with no retransmission control, assured with retransmission control)
    - User-trigger data transfer
        - Assured user-trigger SpacePacket read service
        - Best-effort user-trigger SpacePacket read service

    Note The above "best effort" and "assured" labels correspond to CCSDS SOIS resource-reserved function traffic classes (Applicable Document §7). These should be renamed as in Table §8.1, depending on whether retransmission control is used and whether the SpaceWire-D layer is used.

3. In master-trigger transmission, the master device carries out RMAP Writeor RMAP Readat the necessary timing and transmits data to (or from) the user device.

4. In user-trigger transmission, the master device polls the transmission request flag (or the number of output request packets) of the user device. When a transmission request occurs, the master executes a predetermined RMAP access so that the master device transmits the data to the device.

Table 8.1: Packet service types

SpaceWire-D use	Retransmission control	Corresponding traffic class
No	No	Best-effort traffic class
No	Yes	Assured traffic class
Yes	No	Reserved traffic class
Yes	Yes	Guaranteed traffic class

8.1.2.2 Memory access services

1. Provides functions for accessing user device registers, memory, and so on, such as for device status reading, raw data, and writing of sensor register values.

2. Read and written data should be in raw data format.

3. To simplify implementation on the user-device side, do not provide delivery confirmation and retransmission control. If delivery confirmation or retransmission control is necessary, specify the method in a layer above the subnetwork layer.

8.1.2.3 Time information delivery in synchronization services

1. Provides functions to write time information from time masters to user devices.

2. Time information shall be provided as CCSDS Unsegmented Code (CUC) or as raw data, as defined for each mission.

   Note Raw data formats are project dependent. For example, by following the CUC time field (T-Field) format and deciding bit widths for coarse and fine times according to the project requirements, it is possible to process time information equivalent to that when using CUC, and furthermore likely improve design prospects.

3. For both CUC and raw data formats, detailed specifications such as the bit width, meaning, and resolution of each field should be specified in project documentation and shared among device designers.

8.1.3 Retransmission control using RMAP Reply

In a SpaceWire network, if a link disconnection occurs due to a bit error, there may be a time during which the link is unavailable due to disconnection and initialization procedures taking place. If a link

Table 8.2: Subnetwork services and data formats

Subnetwork service	Data format
Packet service	Service data unit
Memory access service	Raw data
Synchronization service	CUC or raw data

is disconnected while an RMAP command or reply packet is being transmitted, the packet may be interrupted and may reach a destination in a state of being terminated by an EEP. In these cases, the RMAP command is not correctly interpreted on the RMAP target side, RMAP replies returned by the RMAP target may not normally reach the RMAP initiator, and the RMAP transaction may not complete.

When an RMAP transaction fails in a communication sequence requiring reliable data transmission among the SpaceWire-RMAP communication services, the transaction is re-executed by setting the same transaction ID value as that used in the RMAP command of the failed transaction and resending the RMAP command from the master device. Processing of retransmitted RMAP transactions in user devices differs between the cases of RMAP writes and reads, and so the processes are separately defined below.

8.1.3.1 Retransmission control using RMAP Write

1. In a transaction with retransmission control in which the master device performs RMAP writes to the user device, send an RMAP write command packet with reply specified. The transaction ID should be set to a different ID from that used when previously accessing the destination user device via the communication service.

   Reference information To realize this, it is necessary for the master device to store within the device the transaction ID used in the last transaction for each user device and communication service.

2. If the master device cannot receive an RMAP write reply within the specified time (timeout time), create and send an RMAP write command with the same transaction ID as the original RMAP write command packet to reattempt the RMAP write.

3. In user devices, execute the received RMAP command regardless of the transaction ID value. If there are no key, memory address, data length, or other problems, perform a write operation even for a retransmitted RMAP write command (that is, a command packet having the same content as a RMAP write command previously received with the same transaction ID and memory address).

4. If it is necessary to eliminate redundant processing due to retransmission of the same data multiple times, this must be performed by higher-level applications.

   Ex: For example, assume that when writing the TC packet of a CCSDS space packet to a user device with an assured master-trigger write service, the first RMAP write reaches the user device and the received data (TC packet) is written. Then, the written TC packet is received by the upper application, and after checking the validity of the APID and the sequence counter in the TC Packet, the corresponding command is executed if it was valid. If the RMAP write reply for this RMAP write did not return to the master device due to a communication error, the master device would reissue the same transaction ID and perform the RMAP write of the same TC packet again. Upon receiving this RMAP write command, the user device performs a write operation and notifies the upper application of the received TC packet. Since the sequence counter in the TC packet is the same as that of the command executed last time, the upper application receiving the TC packet identifies it as a retransmitted command and does not execute it. Since the RMAP write process itself is executed normally, the RMAP target function in the user device returns an RMAP write reply to the master device (which is the RMAP initiator). If this reply normally returns to the master device, the master device can recognize that the retransmitted RMAP write command was normally processed.

8.1.3.2 Retransmission control using RMAP Read

1. If the master device cannot receive an RMAP write reply within the specified time (timeout time), create and send an RMAP write command with the same transaction ID as the original RMAP write command packet to retry the RMAP write.

2. On the user device side, with respect to the memory area used by the Assured user-trigger SpacePacket read service, do not update the data in the memory area requiring retransmission control until the master device notifies that data collection in the communication sequence has normally completed.

   (a) To cope with the case where an abnormality in the communication procedure continually prevents master device notifications of successfully completed data collection, be sure to specify a timeout period for data retention for retransmission control, or allow resetting the communication sequence management function with a control command.

   (b) When setting timeout times, give the timeout time for each memory area in the ICD and other documents.

   (c) The details of resets by control commands depend on stipulations for the SM&C layer, the device telecommand design, and the system design.

3. On the user device side, when RMAP reads are performed on memory areas used for communication services requiring RMAP read retransmission control, return the data retained in the memory area at that time as an RMAP read reply, regardless of the transaction ID. If an error occurs when reading data, return an RMAP reply with an appropriate reply status set.

4. In master devices, if RMAP reads do not complete normally even after retransmitting more than a specified number of times, notify the upper-level application.

   Note Since this indicates the occurrence of a persistent malfunction, it is necessary to take countermeasures in response to such notifications, such as the higher-level application switching to a redundant system network. Details are system-design dependent.

8.1.3.3 Methods for using transaction IDs

1. 16-bit transaction IDs are divided into two fields, the upper N bits and the lower $16 - N$ bits.

2. The upper N bits are used in the master device to identify the transaction type (target user device, communication service type, application in the master device, etc.).

3. The lower $16 - N$ bits are used to identify transactions within the same transaction type, such as inserting a transaction sequence number.

4. The specific value of N is determined by the project.

8.2 Packet services

Packet services by SpaceWire-RMAP are roughly classified, according to the device determining the timing at which service data units are sent, as master-trigger transmissions (where a master device issues a transmission start request at a predetermined timing) or user-trigger transmissions (where a user device transmits an issuance start request at a predetermined timing).

Master-trigger transmissions can roughly be divided into the two types below, depending on the transmission direction of service data units. In each, the presence or absence of retransmission control can be selected. When retransmission control is not to be performed, it becomes a best-effort class of CCSDS SOIS, or an assured class when retransmission control is performed. When adopting a real-time property guarantee using SpaceWire-D, these correspond to reserved or guaranteed classes, respectively.

1. Master-trigger SpacePacket write services
   (Best effort with no retransmission control, assured with retransmission control)
   Service data unit RMAP Writefrom a master device to a user device
   Usage example:

   - TC packet writes

2. Master-trigger SpacePacket read services
   (Best effort with no retransmission control, assured with retransmission control)
   Service data unit RMAP Readby a master device from a user device
   Usage examples:

   - HK data reads in TM packet format
   - Sensor data reads in TM packet format

   Note The above "best effort" and "assured" labels correspond to CCSDS SOIS resource-reserved function traffic classes (Applicable Document 7). These should be renamed as in Table 8.1, depending on whether retransmission control is used and whether the SpaceWire-D layer is used.

User-trigger transmissions can be roughly divided into the following two types, depending on the presence or absence of retransmission control.

1. Assured-class user-trigger SpacePacket read services
   Service data unit RMAP Readby a master device from a user device. Double handshake communication is performed by polling of transmission requests by RMAP reads, RMAP reads of service data unit (one time), and RMAP write acknowledgements, and provides CCSDS SOIS assured-class services.
   Usage example:

   - HK data reads in TM packet format

2. Best-effort user-trigger SpacePacket read services

   Service data unit RMAP Readby a master device from a user device. Provides CCSDS SOIS best-effort services with confirmations of the number of output packets by polling and subsequent RMAP reads (one or more times) of service data units without retransmission. This allows faster RMAP reads than with the "assured-class user-trigger read service."

   Usage examples:

   - HK data reads in TM packet format
   - Sensor data reads in TM packet format

   Note The above "best effort" and "assured" labels correspond to CCSDS SOIS resource-reserved function traffic classes (Applicable Document 7). These should be renamed as in Table 8.1, depending on whether retransmission control is used and whether the SpaceWire-D layer is used.

When adopting a real-time property guarantee using SpaceWire-D, 1 and 2 are "reserved-class user-trigger read services" and "guaranteed-class user-trigger read services," respectively.

The following chapters stipulate the communication procedures for individual communication services.

8.2.1 Master-trigger write services

In master-trigger write services, service data units such as TC Packetare transmitted from a master device to a user device by RMAP Writetransactions.

1. The user device defines memory areas for master-trigger write services and makes them accessible by RMAP.

2. The master device writes service data units to the memory area of the user device master-trigger write service using RMAP writes, based on requests from upper-level applications.

3. The user device should inform the higher-level application when service data units are written to the memory area for the master-trigger write service.

4. The RMAP target function in the user device returns an RMAP reply when the process of writing service data unit to memory is completed (if a reply is requested in the Instruction field of the RMAP write-command packet).

5. Whether to use retransmission control for master-trigger write services should be specified by system design. To perform retransmission control, follow the stipulations in §8.1.3.

6. If retransmission control is not used, specify in the system design whether to request RMAP replies for RMAP write transactions.

7. Passing service data units between RMAP target functions and higher-level applications in the user device should be atomic in service data units. For example, prevent CPU reads while the RMAP target function is writing service data units to CPU main memory via internal bus access.

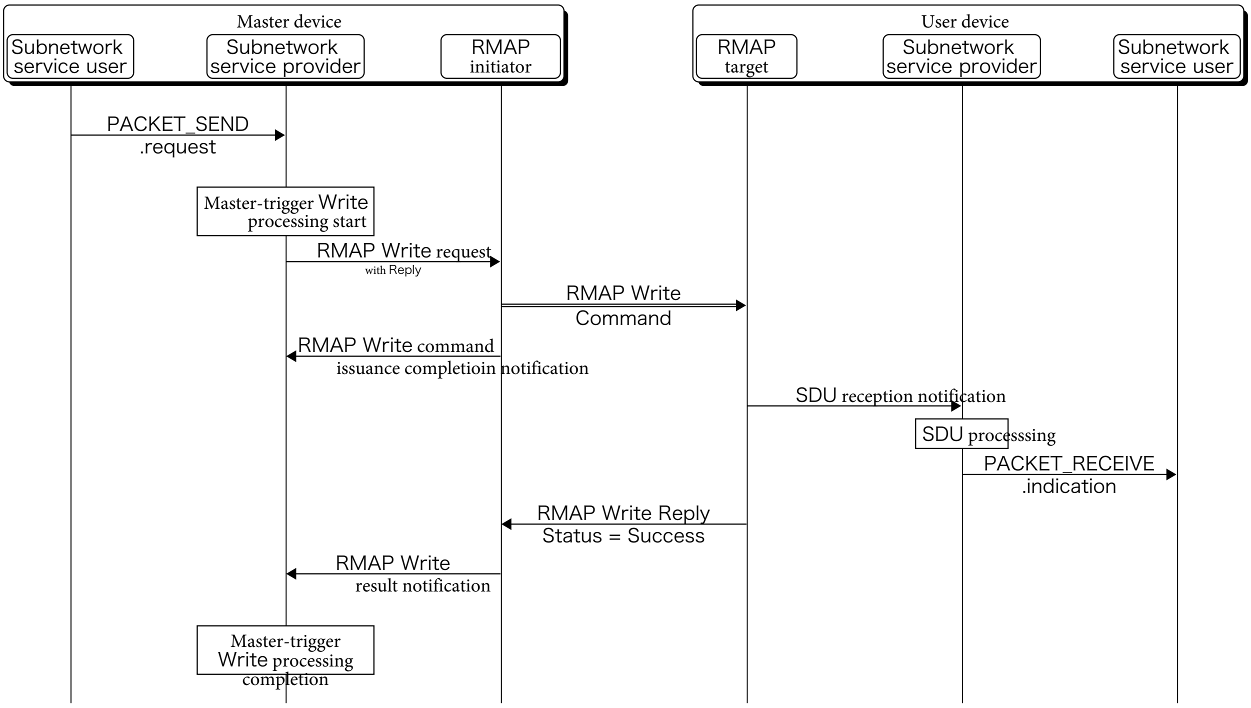

Fig 8.1: RMAP transaction of a master-trigger write service when retransmission control does not occur. Even when retransmission control is not used, the RMAP target side returns an RMAP write reply if one is requested in the Instruction field.

8.2.1.1 Flow of master-trigger write service transactions

Figure 8.1 shows the flow of RMAP transactions in master-trigger write services.

Figure 8.2 shows the flow of RMAP transactions in master-trigger write services with retransmission control in the case where retransmission does not occur.

Figure 8.3 shows the flow of RMAP transactions in master-trigger write services with retransmission control in the case where a first retransmission was successful.

Figure 8.4 shows the flow of transactions in master-trigger write services with retransmission control in the case where no transactions were successful, even after reaching the prescribed maximum number (n) of retransmissions.

8.2.1.2 Parameters for master-trigger write services

Time slot

Time slot in which the master device performs RMAP writes (when using SpaceWire-D).

Memory area

Memory address to which service data units are written in master-trigger write services.

Retransmissions

Whether to perform retransmission control.

Timeout before retransmission

Time until timeout for reply packet reception.

Number of retransmission attempts

How many times retransmission control should be attempted.

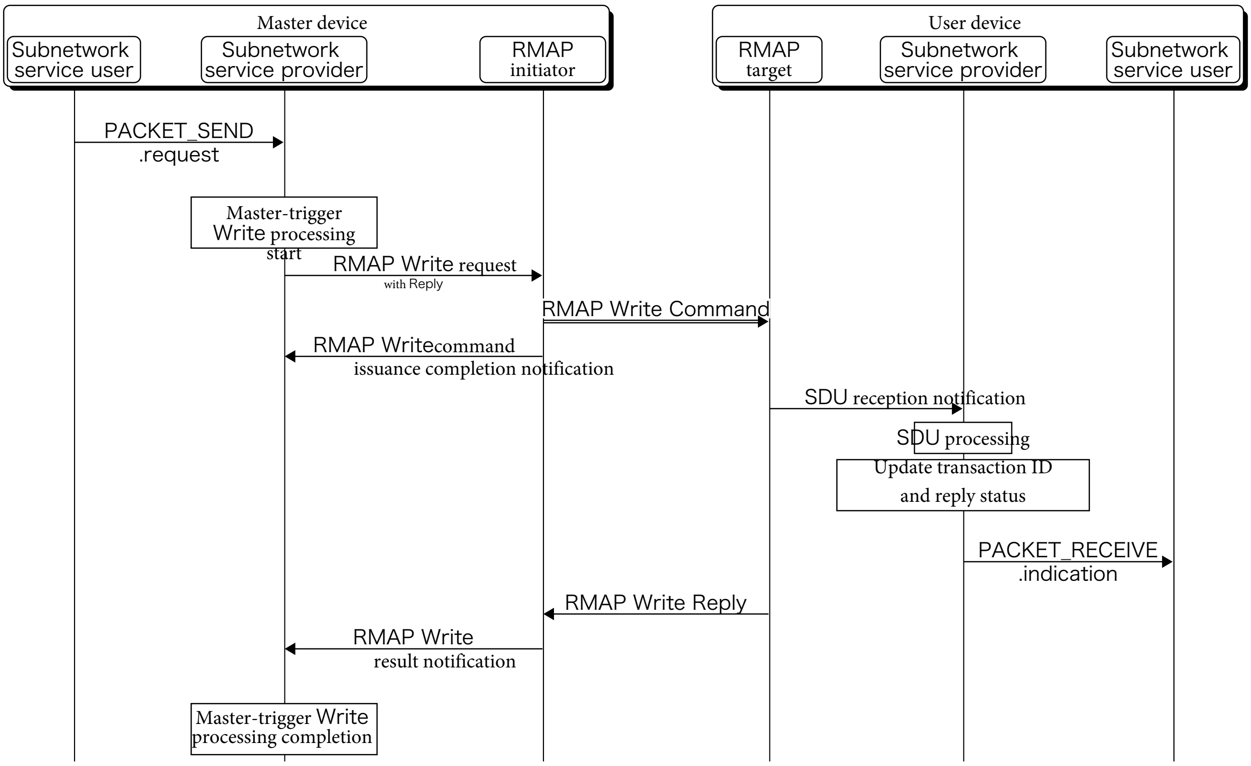

Fig 8.2: RMAP transaction in a master-trigger write services with retransmission control in the case where retransmission does not occur.

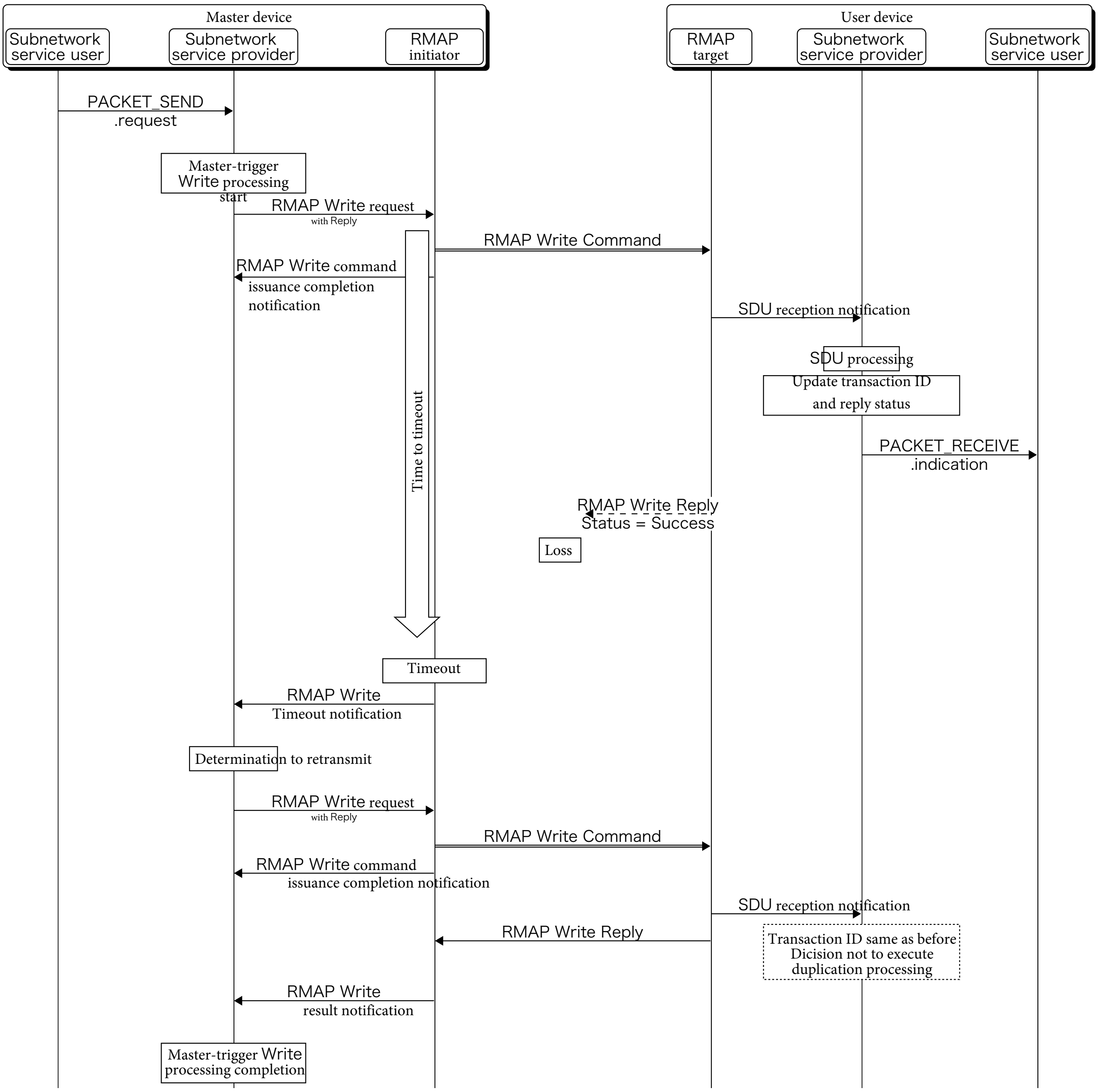

Fig 8.3: RMAP transactions in a master-trigger write service with retransmission control in the case where a first retransmission was successful.

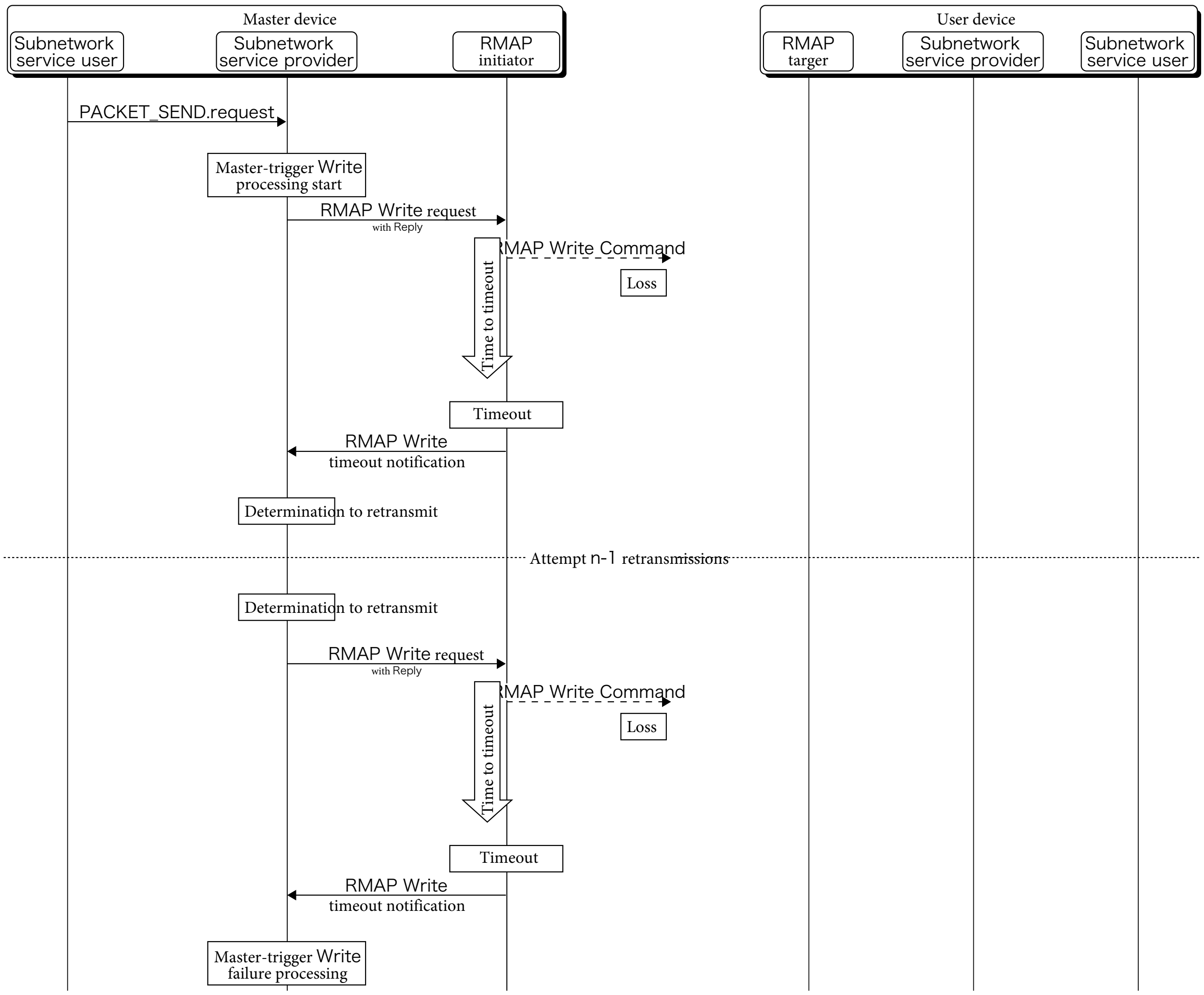

Fig 8.4: Transactions in master-trigger write services with retransmission control in the case where no transactions were successful after reaching the prescribed maximum number (n) of retransmissions.

8.2.2 Master-trigger read services

In master-trigger read services, service data units such as TM Packetare transmitted from a master device to a user device by RMAP Readtransactions.

1. The user device defines memory areas for master-trigger read services and makes them accessible by RMAP.
2. The master device reads service data units from the memory area of the user device’s master-trigger read service using RMAP reads, based on requests from higher-level applications.
3. The upper-level application of the user device prepares service data units in the memory area for the master-trigger read service at a predetermined timing and prepares for RMAP reads by the master device.
4. Whether to use retransmission control for master-trigger read services should be specified by system design. To perform retransmission control, follow the stipulations in §8.1.3.
5. When SpaceWire-D is used, the latency before reply packets are returned is specified. If processing time is required to generate the service data units to be returned by RMAP reads, the user device should generate units before the access from the master device to adhere to delay time specifications.
6. Passing service data units between RMAP target functions and higher-level applications in the user device should be atomic in service data units. For example, prevent CPU modifications while the RMAP target function is reading service data units from CPU main memory via internal bus access.

8.2.2.1 Transaction flow in master-trigger read services

Figure 8.5 shows the flow of RMAP transactions in master-trigger read services without retransmission control.

Figure 8.6 shows the flow of RMAP transactions in master-trigger read services with retransmission control in which no retransmission occurs.

Figure 8.7 shows the flow of RMAP transactions in master-trigger read services with retransmission control in the case where a first retransmission was successful.

Figure 8.8 shows the flow of transactions in master-trigger read services with retransmission control in the case where no transactions were successful, even after reaching the prescribed maximum number (n) of retransmissions.

8.2.2.2 Master-trigger read service parameters

Time slot

Time slot in which the master device performs RMAP reads (when using SpaceWire-D).

Memory area

Memory address from which service data units are read in master-trigger read services.

Retransmissions

Whether to perform retransmission control.

Timeout before retransmission

Time until timeout for reply packet reception.

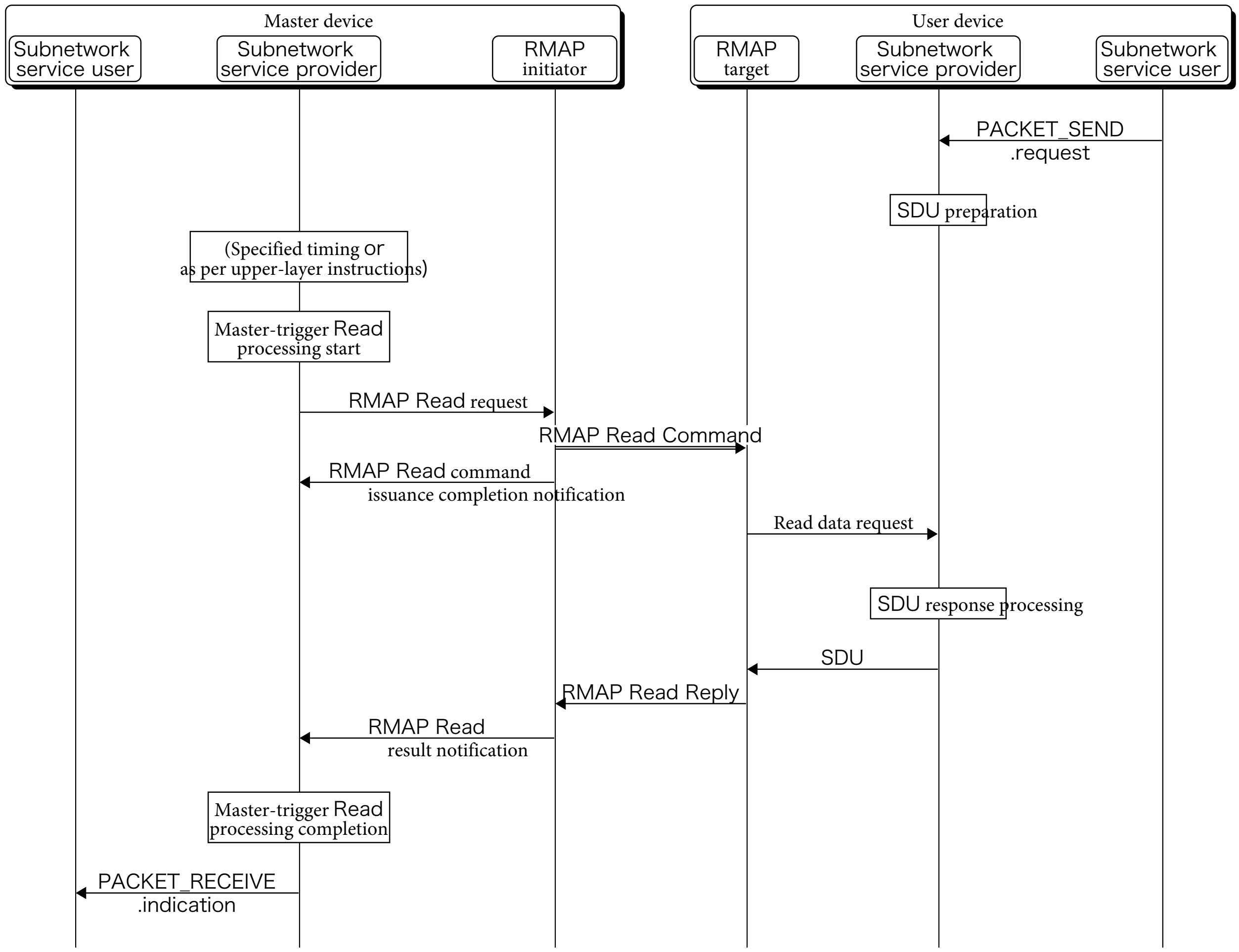

Fig 8.5: RMAP transactions in master-trigger read services without retransmission control.

Number of retransmission attempts

How many times retransmission control should be attempted.

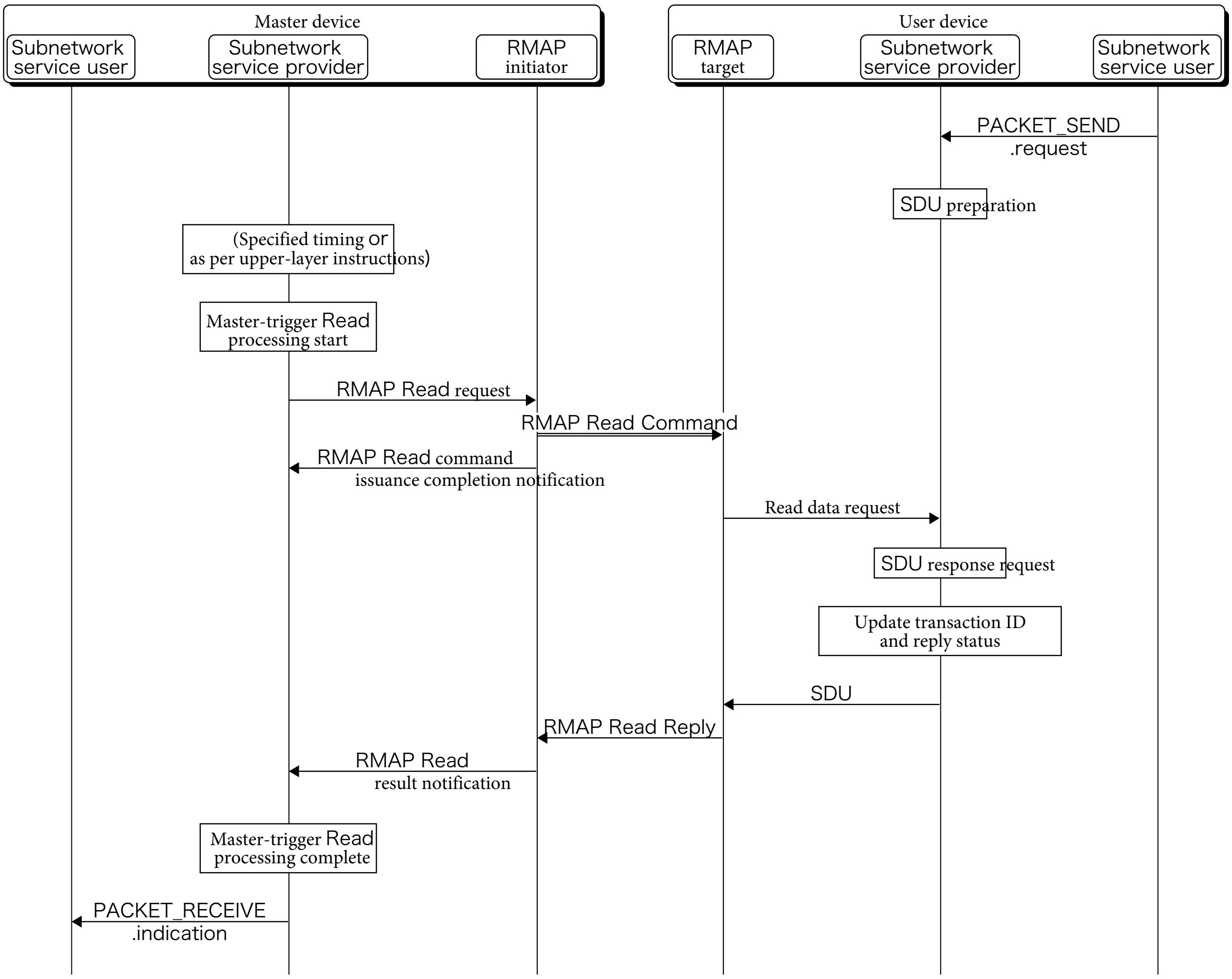

Fig 8.6: RMAP transactions in master-trigger read services with retransmission control in which no retransmission occurs.

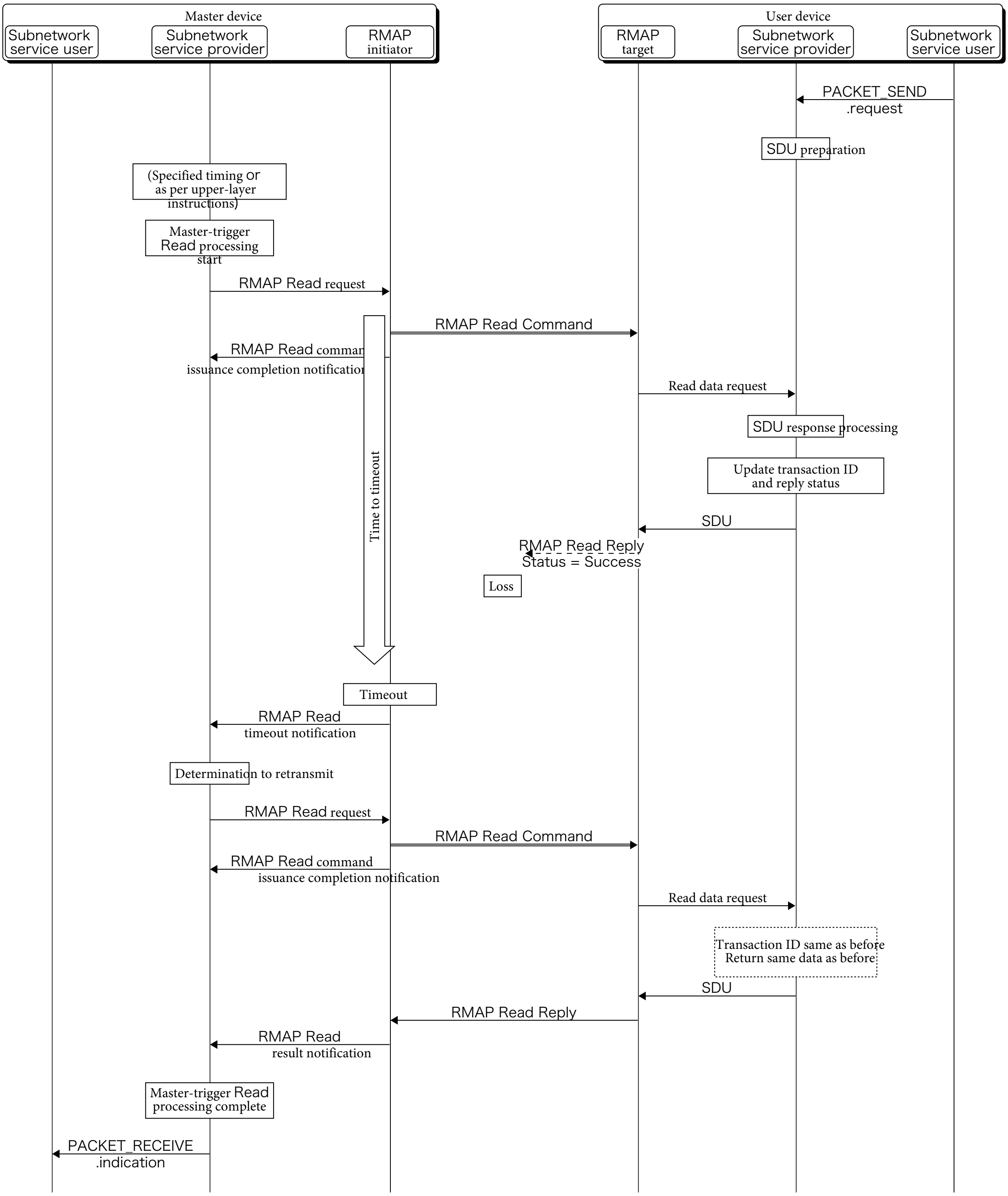

Fig 8.7: RMAP transactions in master-trigger read services with retransmission control in the case where a first retransmission was successful.

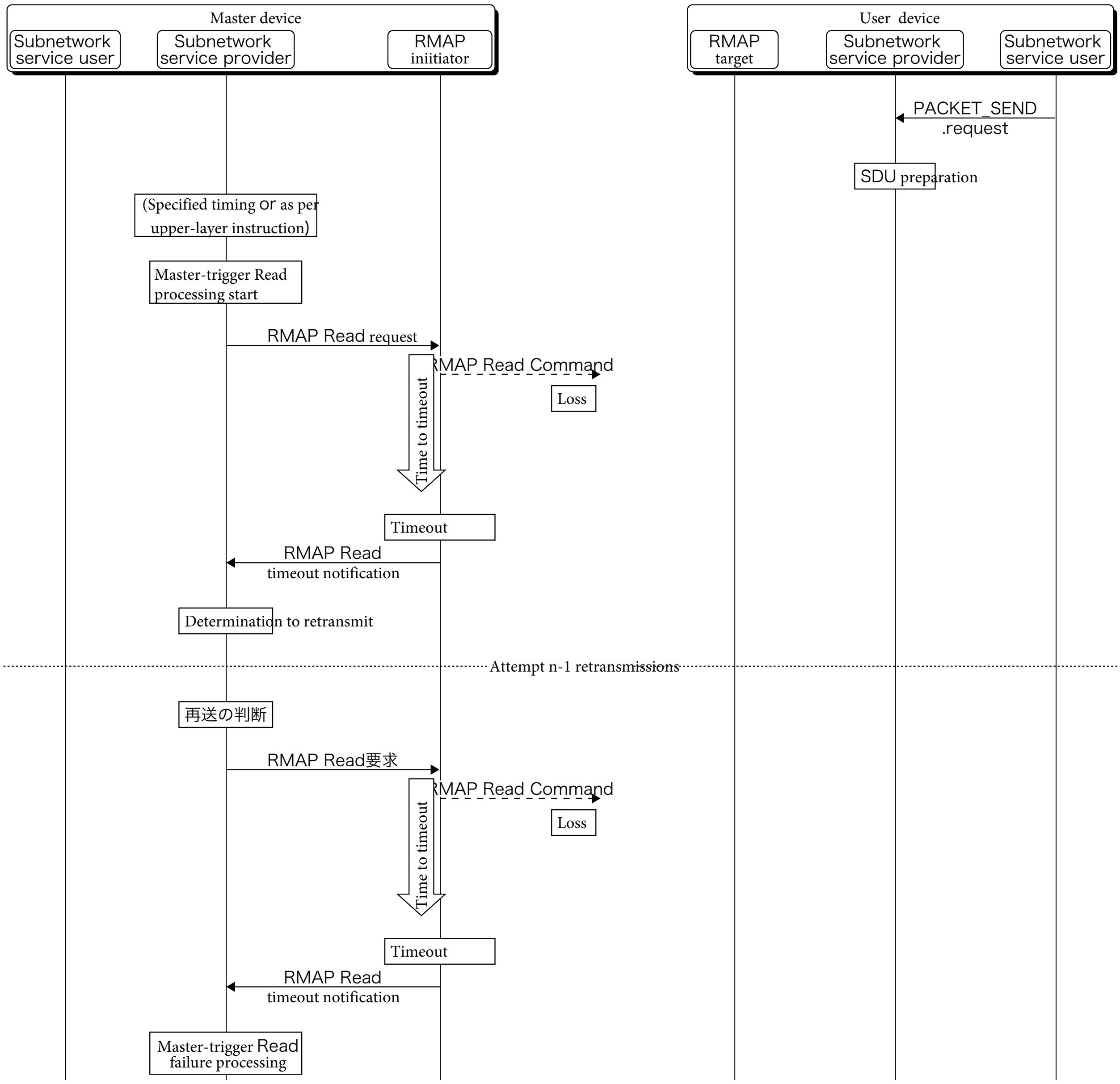

Fig 8.8: RMAP transactions in master-trigger read services with retransmission control in the case where no transactions were successful, even after reaching the prescribed maximum number (n) of retransmissions.

8.2.3 Assured user-trigger SpacePacket read service

With the Assured user-trigger SpacePacket read service, the master device collects data in response to on-demand requests from user devices. By specifying handshakes as a communication sequence between master and user devices, it is possible to reliably transmit data so long as no persistent defect occurs in the SpaceWire network between those devices. In each sequence, RMAP accesses other than polling are executed with replies requested, and if the reply packet does not return within the prescribed time or if an RMAP error status is set in the reply packet, the RMAP access is retransmitted up to the prescribed number of times. When the communication sequence is completed, or if the sequence is not completed even if retrying up to the specified number of times, the result is notified to the higher-level application.

1. Master devices with Assured user-trigger SpacePacket read serviceenabled periodically perform RMAP reads (request polling) of register areas containing related flags on user devices using that service. The master device transitions the internal communication sequence management state machine on the basis of the read flag. The master device transitions internal communication sequence management state machines on basis of the read flags.
2. User devices set request flags indicating transmission start requests.
3. When the request flag is set, the master device collects data from the user device using RMAP reads.
4. Upon completion of data collection, the master device uses an RMAP write to the user device to set an acknowledge flag indicating completion of the collection process.
5. Upon detecting the acknowledge notification, the user device clears the request flag and sets the acknowledge flag.
6. The master device uses polling by RMAP reads to detect that the request flag on the user device side is cleared and that the acknowledge flag is set.
7. The master device clears the acknowledge flag to the user device by RMAP write.
8. The user device completes the Assured user-trigger SpacePacket read servicesequence by clearing the acknowledge flag.
9. The master device uses RMAP read polling to detect that the acknowledge flag on the user device side is cleared.
10. The master device completes the Assured user-trigger SpacePacket read servicesequence.

8.2.3.1 Memory areas used by Assured user-trigger SpacePacket read service

The Assured user-trigger SpacePacket read serviceuses the memory areas described in Table 8.3. User devices using Assured user-trigger SpacePacket read serviceshould allow RMAP access to these memory areas.

Memory areas should fulfill the following stipulations.

1. The request flag area, data size area, and acknowledge flag area are 4 bytes each.
2. The request flag and the acknowledge flag treat 4 bytes as one word.

3. The request flag and the acknowledge flag are defined such that their value (big-endian order) in decimal notation is 1 when the flag is set and 0 when it is cleared.

4. The size of the data area should be larger than the maximum packet length used.

   Note When using SpaceWire-D, specify the maximum packet length so that the transaction is completed within the specified time slot even under expected communication latency.

Table 8.4 shows an example memory map.

Table 8.3: Memory areas used by Assured user-trigger SpacePacket read service

Memory area	Master device access	Overview
Request flag area	Read	Set to 1 when the user device has data for output via Assured user-trigger SpacePacket read service. Clear to 0 upon completion of the output sequence for assured-class data. Master device polls by RMAP reads at a specified time interval.
Data size	Read	Set the data size (packet length in bytes) when a user device has data for Assured user-trigger SpacePacket read service-output. The master device polls by RMAP reads at a specified time interval, using this value as the size when performing RMAP reads on the data area.
Acknowledge flag	Read/Write	An area to store a flag to notify that the master device has completed data area reads for the Assured user-trigger SpacePacket read service. When data reading is completed, the master device writes 1. The user device sets it to 0 when data collection is completed. Like the request flag area, the master device polls by RMAP reads at specified time intervals.
Data	Read	The area where user devices store data (service data units) for output via the Assured user-trigger SpacePacket read service. In the Assured user-trigger SpacePacket read servicesequence, the master device collects stored data using RMAP reads.

Table 8.4: Example Assured user-trigger SpacePacket read servicememory map

Memory area	Start address	End address	Size (bytes)	Notes
Request flag	0xFF80_1000	0xFF80_1003	4	
Data size	0xFF80_1004	0xFF80_1007	4	
Acknowledge flag	0xFF80_1008	0xFF80_100B	4	
Data	0xFF80_2000	0xFF80_5FFF	16k (16×1024)	

8.2.3.2 Assured user-trigger SpacePacket read servicestipulations

1. User devices should allow RMAP access to memory areas to be used with the Assured user-trigger SpacePacket read service(see §8.2.3.1).
2. Allocate the request flag, data size, and acknowledge areas as consecutive memory spaces so that polling from the master device can be performed with a single RMAP read.
3. The master device should maintain a state machine for managing the Assured user-trigger SpacePacket read serviceprocessing status. When providing functionality to simultaneously process multiple Assured user-trigger SpacePacket read serviceinstances, the necessary state machines should be independently maintained.
4. When there are data (service data units) to be output by Assured user-trigger SpacePacket read service, the user device should notify the master side of the request by setting the request flag after placing the data in the data memory area.
5. The master device reads the request status (request flag, data size, and acknowledge flag areas) of the user device by an RMAP read (request polling) all at once. The read request flag and the value of the acknowledge flag are retained in the master device. When the polling RMAP read transaction times out due to a temporary network malfunction, no retransmission (repeated RMAP read) is performed and the request flag is not updated.
6. The master device initiates the Assured user-trigger SpacePacket read servicecollection sequence on the user device for which the request flag is set and collects data.
7. The master device continues polling at the specified period independently of the Assured user-trigger SpacePacket read servicedata collection state machine. Since the state-machine state transition uses the polling result (request flag, changes in the acknowledge flag value), the polling result can be sequentially referenced from the state-machine side.
8. RMAP writes used to set or clear the acknowledge flag should be executed in slots specified in the system design.
9. After RMAP read data collection from the user device is completed, the master device performs acknowledgment for the user device request.
10. The user device detects that an acknowledge has been written and clears the request flag.
11. If timeouts, RMAP status errors, or similar events occur during data collection or acknowledgment communication sequences, attempt retransmission control up to the prescribed number of times. If user device data collection remains incomplete after doing so, notify the higher-layer application with error information. (Retransmission is not performed, due to stipulations that polling should be performed periodically.)
12. In the system design, it is necessary to specify time slots that may be polled, time slots where data area reads may be performed, and time slots in which completion notices are written in the acknowledge area.

13. The master device should be able to manage memory Assured user-trigger SpacePacket read serviceareas for each user device using that service, with at least one area for each device.

Recommendation Regarding the above item 13, it is recommended to define one Assured user-trigger SpacePacket read servicememory area for each user device and to use the same area for multiple data types, because doing so makes master device design simpler. In particular, when multiple Assured user-trigger SpacePacket read servicememory areas are defined for each user device, the processing load from periodic RMAP read request status polling alone increases linearly.

8.2.3.3 Assured user-trigger SpacePacket read servicetransactions

Figures 8.9 and 8.10 show the Assured user-trigger SpacePacket read servicecommunications sequence.

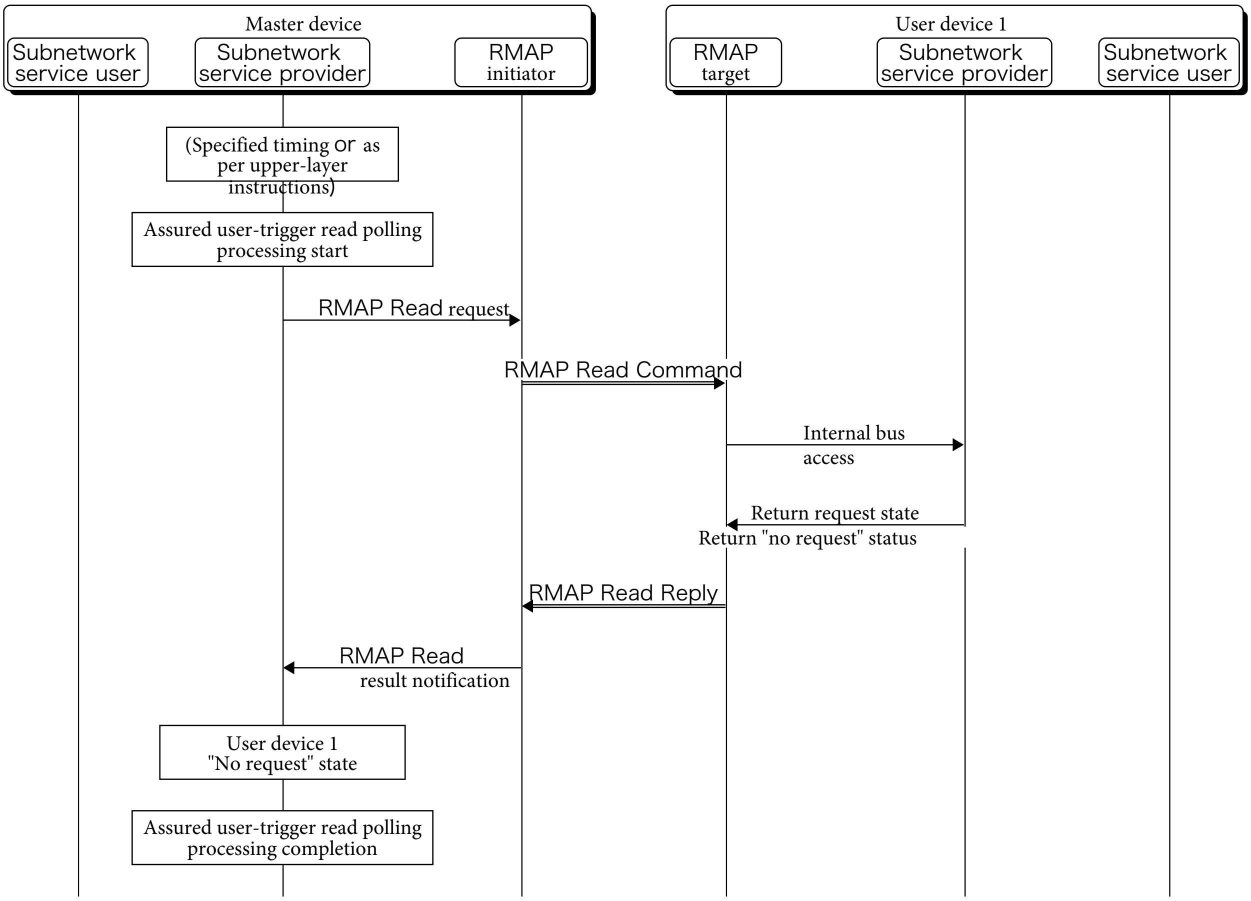

Fig 8.9: Assured user-trigger SpacePacket read servicecommunications sequence with no data output requests from user devices.

Figures 8.12, 8.13, and 8.14 show state-transition diagrams for master and user devices in the Assured user-trigger SpacePacket read service.

8.2.3.4 Assured user-trigger SpacePacket read serviceparameters

Memory area

Address of memory area needed by the Assured user-trigger SpacePacket read service.

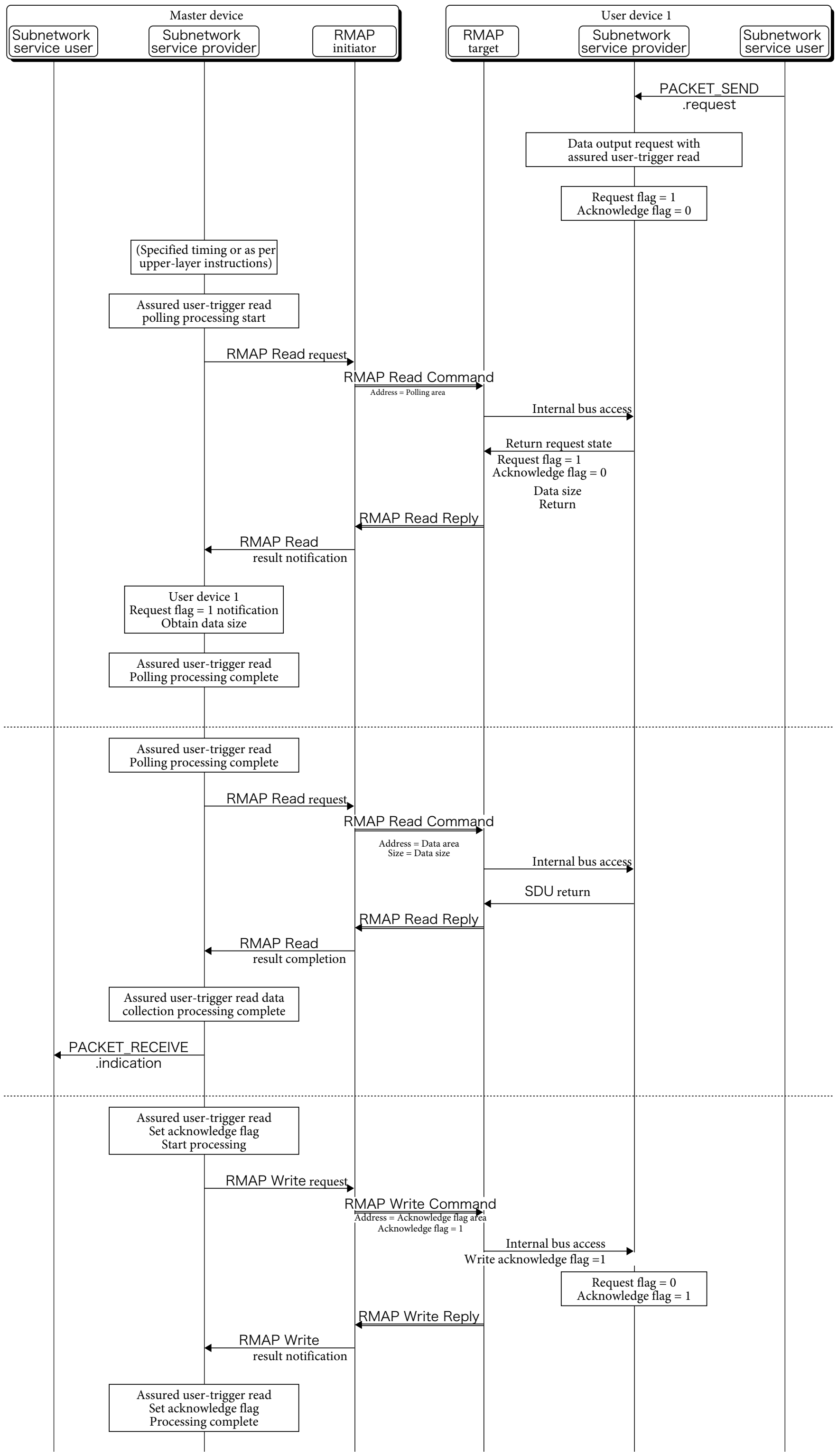

Fig 8.10: Assured user-trigger SpacePacket read servicecommunications sequence. Continues in Fig. 8.11.

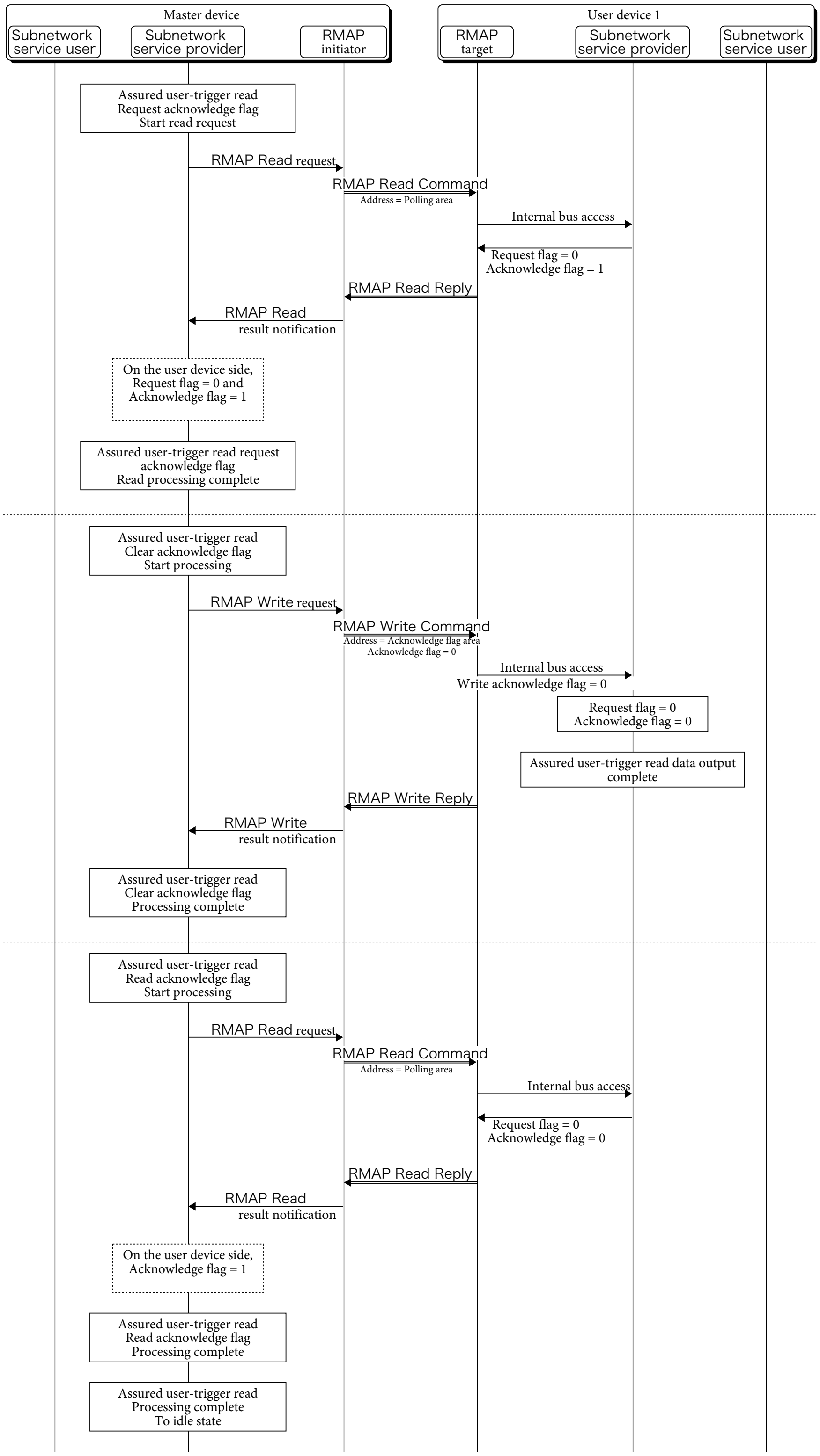

Fig 8.11: Assured user-trigger SpacePacket read servicecommunications sequence. (Continues from Fig. 8.10)

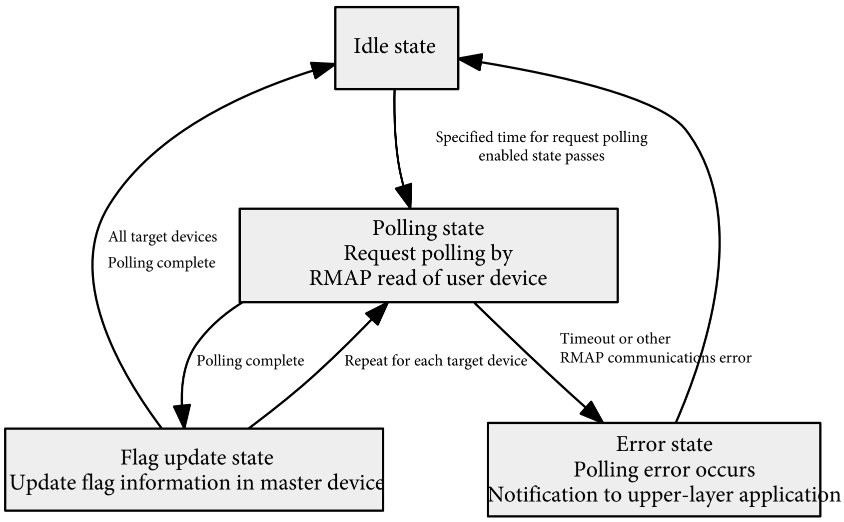

Fig 8.12: State-transition diagram for master device request polling in the Assured user-trigger SpacePacket read service.

Polling timing

Time slot and period for polling the request flag area with RMAP reads.

Acknowledge flag RMAP write timing

The time slot and period at which the master device sets or clears the acknowledge flag area with RMAP writes in the Assured user-trigger SpacePacket read servicesequence.

Number of retransmission attempts

Maximum number of times to try retransmission control. When this differs by memory area, specify the value for each area.

Time until timeout

Time threshold for detecting master device timeouts.

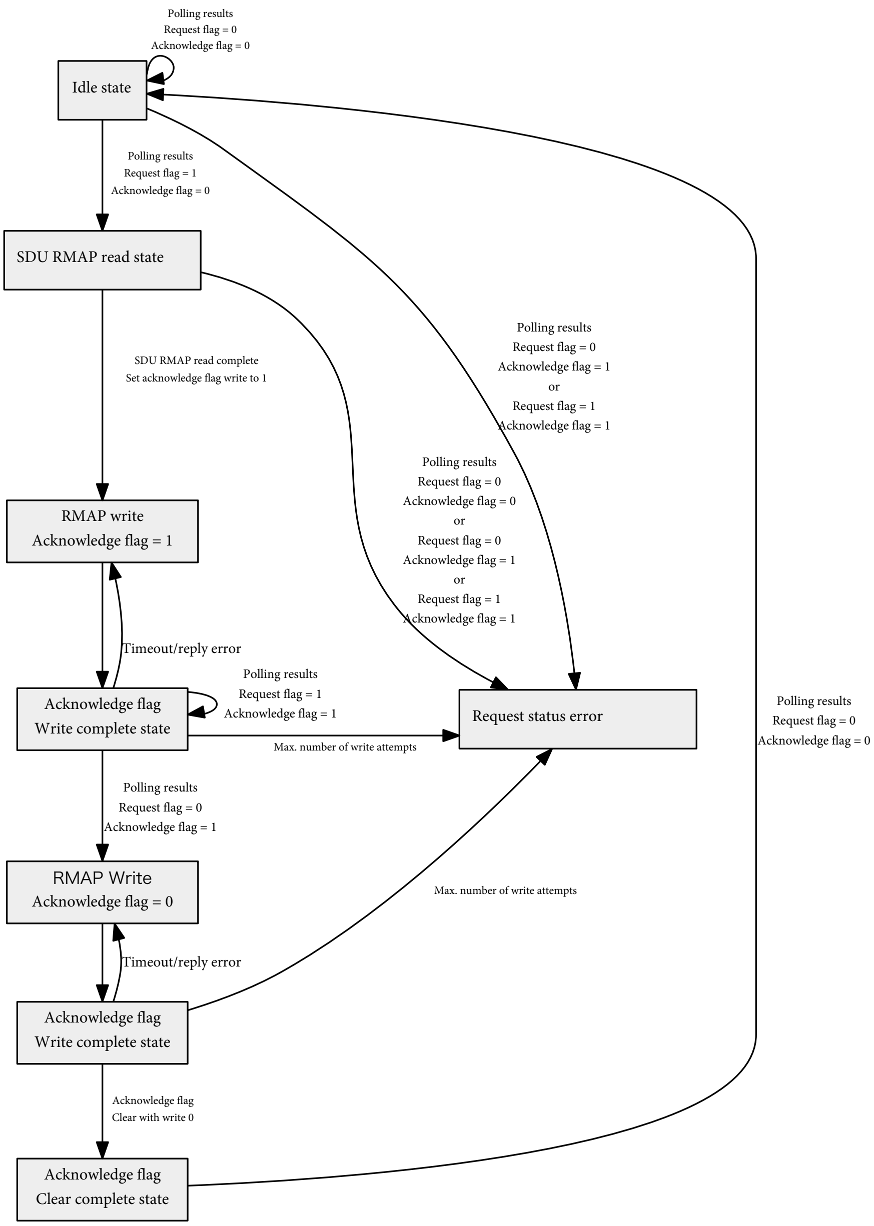

Fig 8.13: State-transition diagram for master devices in the Assured user-trigger SpacePacket read service.

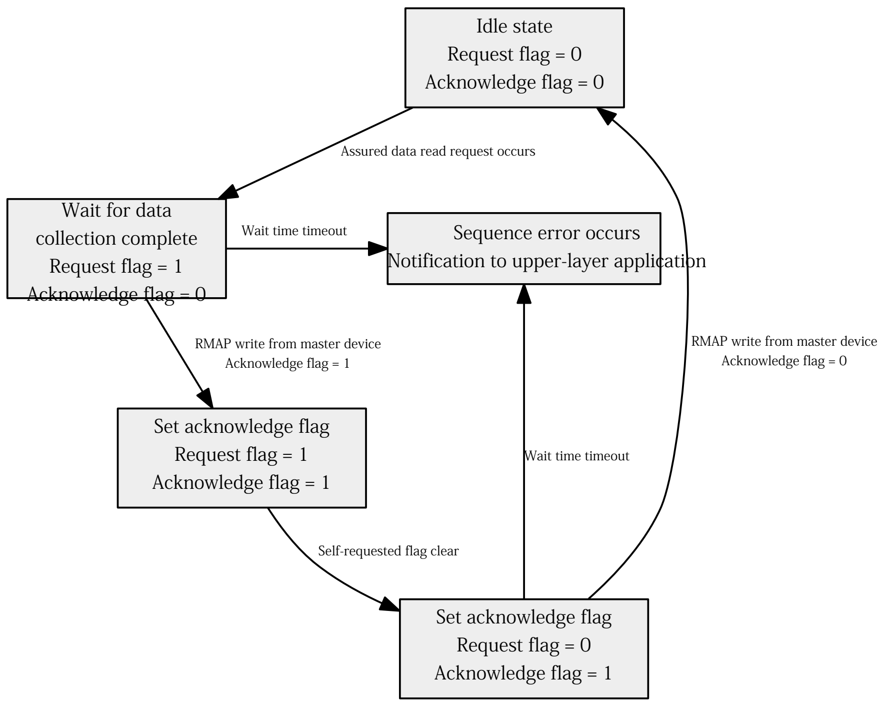

Fig 8.14: State-transition diagram for user devices in the Assured user-trigger SpacePacket read service.

8.2.4 Best-effort user-trigger SpacePacket read service

The Best-effort user-trigger SpacePacket read serviceprovides higher data transmission rates by simplifying the communication sequence used under the Assured user-trigger SpacePacket read service. This service can be used for outputting TM packets of mission sensor observation data and memory dump data in TM packet format generated upon TC packet receipt. To simplify the communication sequence, the data size used when the master device accesses the user device data area takes a fixed value, as specified in the system design. When variable-length data are output within a range not exceeding the maximum value, define an idle packet, pad the end, and provide fixed-size replies for data collection from the master device. Alternatively, the data size of the RMAP read reply can be made shorter than that specified by the RMAP read command, and the data can be returned without padding.

By adopting this simplified communication sequence, it is easier to implement higher-level applications for master and user devices in hardware than under the Assured user-trigger SpacePacket read service. Data transmission speeds can thus be improved over those of software-based communication sequences.

The following is an overview of communications procedures under the Best-effort user-trigger SpacePacket read service.

1. Set to the the output request packet number areathe number of service data units that the user device wishes to output via the Best-effort user-trigger SpacePacket read service. When the telemetry output rate of a given user device is fixed, set the fixed value as specified in the system design.

2. The master device reads the output request packet number areafrom the user device by polling using RMAP reads.

3. If the output request packet number areais not 0, the master device reads service data units from the user device using an RMAP read (in a single read of the specified size). Every time a packet is read out, the output request packet number counter managed for each user device in the master device is decremented by one. The number of packets collected from a user device within the specified time is specified by the system design.

4. Every time the Best-effort user-trigger SpacePacket read servicedata area is accessed, the user device decrements the internally held output request packet number counter by one. Depending on the number of user device collections within the time allocated by the system design and the circumstances of the master device, it is possible that not all packets requested by the user device will be collected within the specified time.

5. Retransmission is not performed even if a timeout or an RMAP reply status error occurs when the master device collects data by RMAP read transactions. (Even in the case of a timeout or status error, the number of output request packets managed on the master device side is decremented by 1.)

6. When the output request packet number counter managed by the master device reaches 0, the master device completes the assured data collection servicesequence. Even when its value does not become 0 within the specified time, the counter in the master device is updated with the value of the output request packet number counter obtained by the next polling.

8.2.4.1 Memory areas used in the Best-effort user-trigger SpacePacket read service

The Best-effort user-trigger SpacePacket read serviceuses the memory areas listed in Table 8.5. User devices that use the Best-effort user-trigger SpacePacket read serviceshould implement these memory areas so that they can be accessed by RMAP. Table 8.6 shows an example Best-effort user-trigger SpacePacket read servicememory map.

Table 8.5: Memory areas used in the Best-effort user-trigger SpacePacket read service

Memory area	Master device access	Overview
Output request packet number	Read	Number of packets to be output when there are data that the user device wishes to output via the Best-effort user-trigger SpacePacket read service. Set to 0 when data output for the requested amount is completed. The master device polls by RMAP reads at a specified time interval.
Data	Read	Stores data (service data units) that the user device wishes to output via the Best-effort user-trigger SpacePacket read service. In the Best-effort user-trigger SpacePacket read servicesequence, the master device performs RMAP reads to collect the stored data.

Table 8.6: Example Best-effort user-trigger SpacePacket read servicememory map.

Memory area	Start address	End address	Size (bytes)	Notes
Output request packet count	0xFF80_1000	0xFF80_1003	4	
Data	0xFF80_6000	0xFF80_9FFF	8k (8×1024)	

8.2.4.2 Best-effort user-trigger SpacePacket read servicestipulations

1. User devices should make the memory area (8.2.4.1) used by the Best-effort user-trigger SpacePacket read service) RMAP accessible.
2. To manage the Best-effort user-trigger SpacePacket read serviceprocessing status, the master device should hold an output request packet number counter for each user device using the service.
3. When a user device has data to be output by the Best-effort user-trigger SpacePacket read service, the device should place the data in the data area, then notify the master with a data output request by setting a value in the output request packet number area.
4. The master device polls the output request packet number area of the user device by RMAP reads at the specified time, and starts the Best-effort user-trigger SpacePacket read servicecollection sequence if a nonzero value is set. When the polling RMAP read transaction times out due to a temporary network malfunction, retransmission (a repeated RMAP read) is not performed and the output request packet number counter is not updated.
5. For devices with a nonzero output request packet number counter, the master device collects data from the user device using an RMAP read at the specified time slot (in a single read of the specified

size). Each read packet reduces the output request packet number counter, managed by the user device in the master device, by one.

6. Every time there is an access to the Best-effort user-trigger SpacePacket read servicedata area, the user device returns the data, sets data in the memory area for the next RMAP read, and prepares to return data for the next RMAP read from the master device. Also, the internally held output request packet number counter is decremented by 1.
7. Retransmission is not performed even if a timeout or an RMAP reply status error occurs when the master device collects data by RMAP read transactions. (Even in the case of a timeout or status error, the number of output request packets managed on the master device side is decremented by 1.)
8. When the output request packet number counter managed by the master device reaches 0, the master device completes the assured data collection servicesequence. Even when its value does not become 0 within the specified time, the counter in the master device is updated with the value of the output request packet number counter obtained by the next polling.
9. The master device continues polling (RMAP reads) of the output request packet number area at the specified period independently of data collection by the Best-effort user-trigger SpacePacket read serviceand updates the output request packet number counter in the master device.
10. The system design should specify time slots for polling of output request packet number counters and time slots for reading from the data area.
11. The system design should specify the RMAP read size (the maximum size of a TM packet read with one RMAP read) for data collection in the Best-effort user-trigger SpacePacket read service.
12. The system design should specify the number of packets and packet size that each user device can output within the specified time. In particular, when calculating the data transmission rate, keep in mind that, depending on the number of user devices collected within the allocated time and the situation of the master device, not all packets requested by user devices may be collected within the specified time.
13. Keep in mind that the Best-effort user-trigger SpacePacket read servicedoes not perform retransmission due to timeouts or reply status errors from RMAP reads in the data area. (Once the user device overwrites data returned from a user device as an RMAP read reply, this service does not perform retransmission.)
14. If the service data unit to be output is shorter than the size of one RMAP read in the data area, the user device may return data of a smaller number of bytes than required in an RMAP read command packet (in this case, set the data length field in the RMAP reply as appropriate).
15. The master device should be able to retrieve stored service data units even from RMAP reply packets with a shorter data length than requested by RMAP reads.
16. The size of the data area should be larger than the maximum packet length to be used.

    Note When using SpaceWire-D, specify the maximum packet length so that the transaction is completed within the specified time slot even when considering communication latency.

8.2.4.3 Best-effort user-trigger SpacePacket read servicetransactions

Figures 8.15 and 8.16 show the communication sequence used by the Best-effort user-trigger SpacePacket read service.

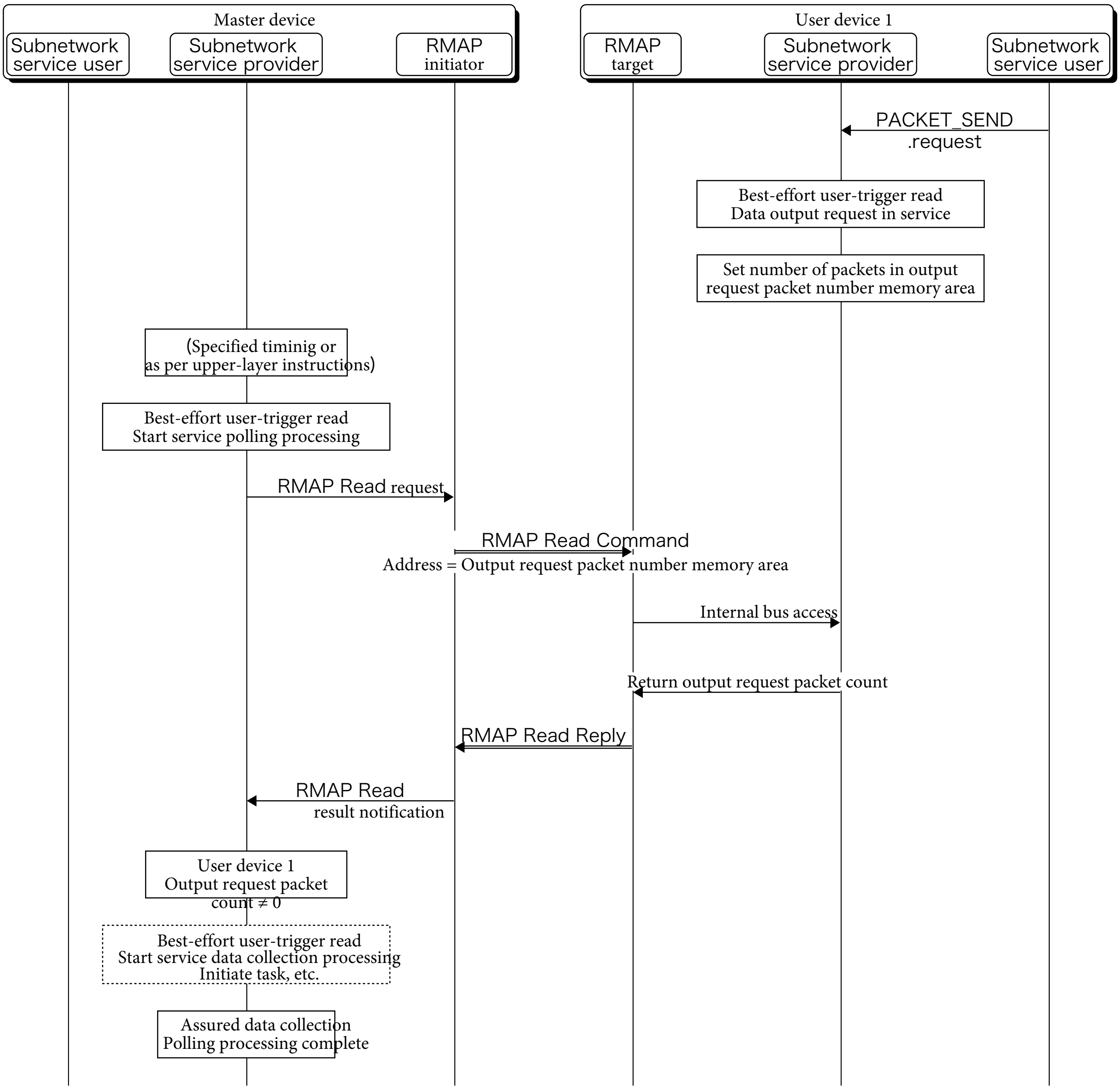

Fig 8.15: Best-effort user-trigger SpacePacket read servicecommunication sequence for polling of the output request packet number area.

8.2.4.4 Best-effort user-trigger SpacePacket read serviceparameters

Memory area

Address of the memory area required by the Best-effort user-trigger SpacePacket read service.

Polling timing

Time slot and period for polling the output request packet number area with RMAP reads.

Data collection timing

The time slot at which RMAP reads can read the data area, the number of transactions in a single time slot, and the shortest time interval between successive RMAP read commands (the time interval that can be processed by the user device RMAP target function, or the time interval that does not affect the timeout time even if the process is executed sequentially).

Time to timeout

Time to detect master device timeouts.

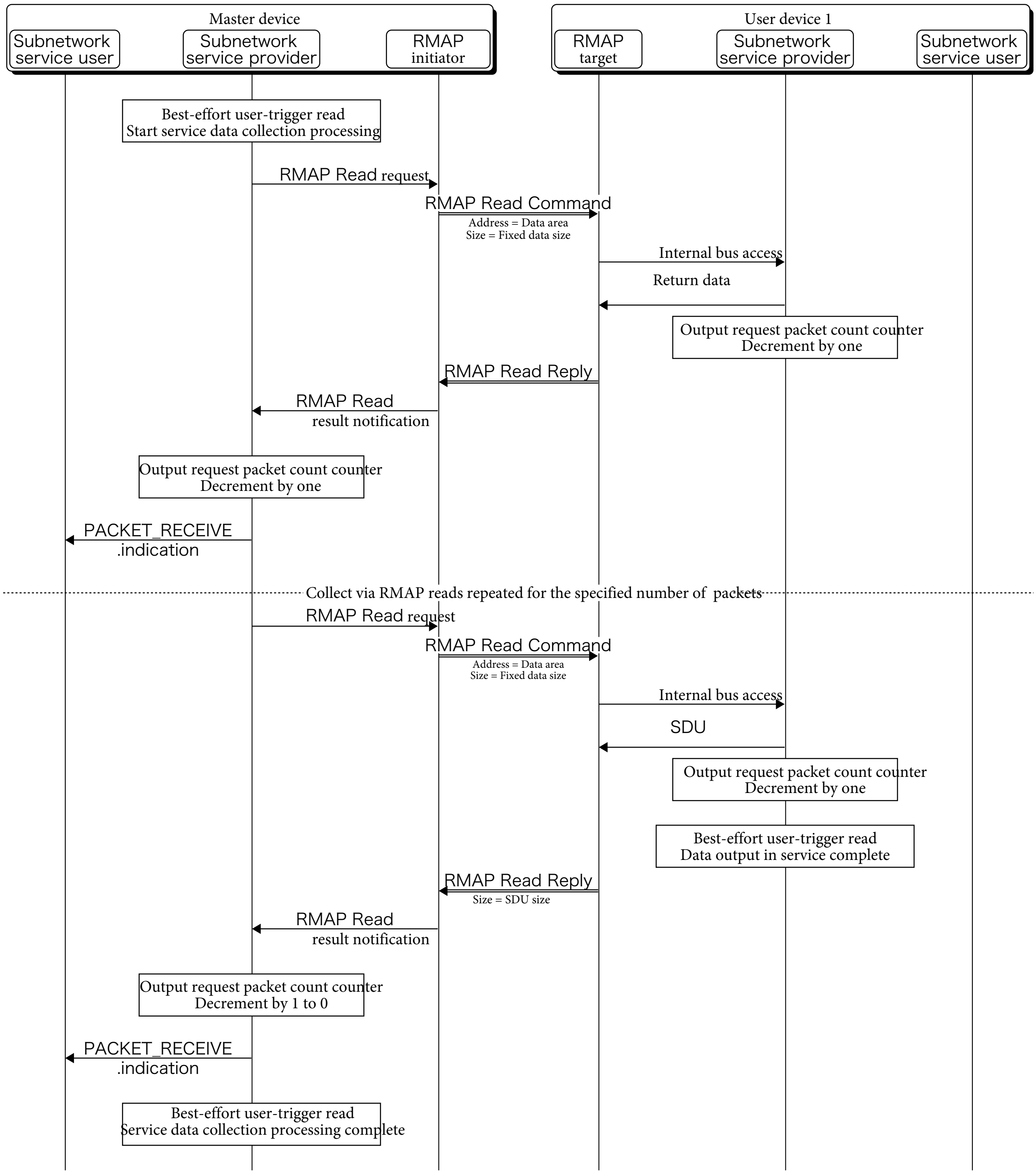

Fig 8.16: Best-effort user-trigger SpacePacket read servicecommunication sequence for data collection.

8.3 Memory access services

Since memory access services are provided by SpaceWire-RMAP, there are no special stipulations for communication procedures.

8.3.1 Memory access services by RMAP reads and writes

Consider the following points as general concerns when accessing memory with RMAP reads or writes.

1. If retransmissions are necessary, they should be performed in higher layers (the SM&C layer etc.).
2. As a rule, memory access services handle raw data.
3. If the data to be handled are service data units in formats defined as CCSDS space packets, transfer frames, virtual channel data units, and so on, use one of the packet service communication services stipulated in §8.2.
4. Memory maps that user devices maintain and memory areas that user devices expose to the RMAP initiator device in the RMAP memory space should be shared in ICDs between the master and user device at the system design stage.
5. Since the data handled by this service are raw data and explicit formats such as CCSDS space packets, transfer frames, and virtual channel data units are not applied, to prevent inconsistency in understanding, ICDs and other documentation should explicitly specify the meaning of data formats read and written in the memory area and the values they can take.

8.3.1.1 Parameters for memory access services using RMAP reads and writes

Memory area and access direction

The address map of the RMAP-accessed memory area from the master device and the readable and writable attributes of each memory area.

Access timing

The timing of RMAP reads and writes (when adopting SpaceWire-D, specify a time slot).

Time to timeout

Time to detect master device timeouts.

8.4 Time information distribution in synchronization services

As a SpaceWire-RMAP synchronization service, SpaceWire Timecode provides the Master-trigger Time write services, which distributes information of upper time digits that cannot be synchronized from the time master device to the user device at specified intervals by RMAP writes.

8.4.1 Master-trigger Time write services

8.4.1.1 Master-trigger Time write servicesstipulations

1. The master device should write time information to the time information distribution memory area of the user device at the specified time with RMAP writes.
2. The time information to be written shall be the time information when the next Timecode is 0.
3. The format of time information shall be CCSDS unsegmented code (CUC) or raw data, as adjusted and defined during system design.
4. If retransmission control is necessary for time synchronization in the system design, upon receiving a Reply packet including an error, or upon receiving a wait timeout for a Reply packet, attempt retransmission up to the specified number of times on the master side. Apply the stipulations of § 8.1.3 for retransmission control.
5. On the user device side, update internal time information with the written time information when the next Timecode is 0.

Recommendation In this design standard, retransmission control is not recommended for the Master-trigger Time write services. This is because retransmission control in this design standard targets transient communication errors, but even if the time distribution fails transiently, the user device can extrapolate times using a local clock.

Notes When retransmission control is used for time distribution, note that RMAP writes with reply generally have higher network occupancy rates and impose increased processing loads on the master device, compared with when replies are not requested, due to the need for processing (transaction state management, reply packet interpretation, etc.). In many cases time distribution to multiple devices is required to be simultaneous (for example, in the same time slot) to the extent possible, and deterioration of real-time performance may be a problem if there are many user devices for time distribution. Consideration of this is required during system design.

8.4.1.2 Master-trigger Time write servicestransactions

Figure 8.17 shows the flow of RMAP transactions with no retransmission control.

8.4.1.3 Master-trigger Time write servicesparameters

Distribution period

Period for time distribution

Time data format

Whether upper time digits are represented in CUC or raw data format

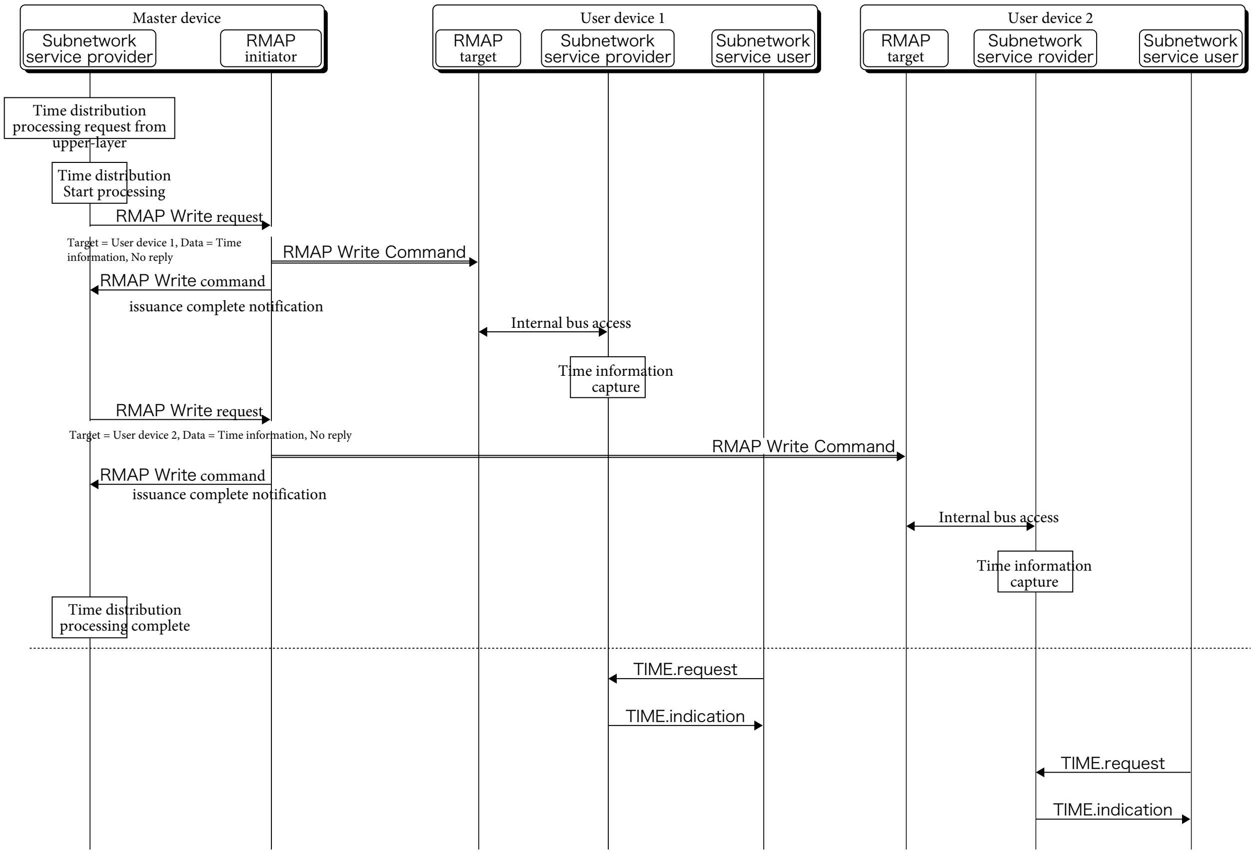

Fig 8.17: Time distribution transaction with no reply.

Memory area

Memory address to which upper time digits are written

Retransmissions

Whether retransmissions are performed

Timeout until retransmission

Time to wait for reply packet reception before timeout

Number of retransmission attempts

The number of times that retransmission control should be attempted when retransmission control is performed

8.5 Reference: Correspondence with ASTRO-H/Hisaki communication service definitions

Table 8.7 shows correspondences with the communication service in the scientific satellite ASTRO-H/Hisaki, which became the basis of communication service definitions for this design standard using SpaceWire-RMAP.

Table 8.7: Correspondence with ASTRO-H/Hisaki communication services.

SpaceWire design standard Communication service	in A-H/Hisaki Corresponding service
Required/supplementary HK collection	Memory access service (RMAP reads)
Master-trigger collection	Master-trigger read service
Guaranteed user-trigger collection	Guaranteed user-trigger read communication service *
Non-guaranteed user-trigger collection	Reserved user-trigger read communication service *
Command delivery	Master-trigger write service
Time delivery	Master-trigger write service
SMU/DHFS general-use RMAP write command	Memory access service (RMAP write)

* Since this adopts a real-time guarantee by time slot division equivalent to that of SpaceWire-D, this is of the guaranteed/reserved traffic type, not the assured/best-effort type described in this design standard. See Table 8.1 for details.

Chapter 9

Subnetwork services using SpaceWire-PTP

9.1 Overview

1. Subnetwork services using SpaceWire-PTP can provide the following services.

   (a) Packet services

   (b) Time information distribution required by synchronization services

2. Subnetwork services using SpaceWire-PTP should not implement send confirmations, resend control, or segmentation. When these features are required, adopt subnetwork services using SpaceWire-R, or separately stipulate a higher (e.g., SM&C) layer.

3. In cases where real-time data transfer must be guaranteed, or when maximal levels of packet transfer latency must be guaranteed, use TimeCode for time division control (§4) as a lower layer.

4. Do not provide a memory access service. When one is required, use a SpaceWire-RMAP subnetwork service, or stipulate an upper-level communication procedure.

5. When designing the topology and link rate of a SpaceWire network, reserve sufficient communication paths and bandwidth to transmit the amount of CCSDS space packet data sent from each device.

9.2 Packet service

Packet services using SpaceWire-PTP encapsulate CCSDS space packets into SpaceWire packets based on the SpaceWire-PTP specification and transmit them between nodes as SpaceWire packets.

9.2.1 Packet service stipulations

1. Transmitted data formats must be CCSDS space packets.

2. Devices that send CCSDS space packets should provide requesting upper-level applications with specified address information (SpaceWire path destinations and logical addresses), encapsulate them as SpaceWire packets, and send them.

   Note Use the similarly named “CCSDS Packet Transfer Services Stipulated by SpaceWire-PTP” (§5.2 in Applicable Document 4) to realize packet services for CCSDS SOIS subnetwork services.

3. Devices that receive CCSDS space packets should perform the operation specified by the upper layer.

Ex: A device receiving a TC packet from a satellite control device executes the TC packet.

Ex: Mass memory receiving a TM packet from a sensor device writes the TM packet into the recording area.

9.2.2 Packet service parameters

Data generation amounts and destinations

Packet length, frequency, and destination of CCSDS space packets generated or transmitted by each device

CCSDS space packet maximum length

Maximum length of CCSDS space packets sent between devices

CCSDS space packet send timing

(When adopting TimeCode-based time division control (§4)) Time slots of each device-destination pair for sending packets

9.3 Time information distribution in synchronization services

The synchronization service using SpaceWire-PTP provides time information that cannot be synchronized with SpaceWire TimeCodes from a time master device to each device.

9.3.1 Stipulations for time information delivery

1. Time information must be in CCSDS Unsegmented Code (CUC) format.
2. Transmitted CUC information should be time information for when the next SpaceWire TimeCode is 0.
3. Nodes receiving CUC information should use the received time information to update internal time information when the SpaceWire TimeCode is 0 or at an internally defined time.
4. SpaceWire specification makes simultaneous delivery to multiple devices impossible. Time master devices should generate an independent SpaceWire packet for each device to be time-delivered, and send it by SpaceWire-PTP.

9.3.2 Time information delivery parameters

Target devices for sending time information

Those devices connected to a SpaceWire network that will receive time information.

Timing of time information transmission to each device

(When adopting TimeCode-based time division control ((§4)) Period and phase (time slot) to transmit CUC data to each target device.

Chapter 10
Subnetwork services using SpaceWire-R

10.1 Overview

1. Subnetwork services using SpaceWire-R can provide the following services.
   (a) Packet services
   (b) Time information distribution required by synchronization services
2. In subnetwork services using SpaceWire-R, delivery confirmation, retransmission control, and segmentation are automatically performed by functions provided by the SpaceWire-R protocol.
3. When guaranteeing real-time performance of data transmission or worst-case guarantees of packet transfer latency are required, use TimeCode-based time division control (§4) as a lower layer.
4. Do not provide a memory access service. When one is required, use a SpaceWire-RMAP-based subnetwork service, or stipulate the communication procedure in an upper layer.
5. SpaceWire-R communications require many more parameter settings than are required for SpaceWire-RMAP or SpaceWire-PTP. To improve the reliability and reduce the number of processes for design, verification, and operation, consider the items presented in §10.2.2 and §10.3.2 during the system design stage, and ensure sharing of ICD and other documentation to prevent misunderstandings.

Recommendation Topologies that achieve (or nearly achieve) peer-to-peer connections between sensor devices and mass memory, mass memory and downlink equipment, and similar, are recommended in communications using the SpaceWire-R protocol. This is because in SpaceWire-R-based communication, segmentation, acknowledgements, and retransmission control complicate bandwidth utilization efficiency in the SpaceWire layer and increase the complexity of calculations in packet transmission latency upon retransmission. Performing design verification using the SpaceWire-R protocol over a large SpaceWire network composed of many SpaceWire routers will likely require many steps. In topologies that are nearly peer-to-peer, performance estimations are relatively easy, and this design standard serves as a handbook that presents concrete performance parameters (Related Document §??).

10.2 Packet services

Packet services using SpaceWire-R transmit service data units as SpaceWire packets between nodes based on SpaceWire-R stipulations.

10.2.1 Packet service stipulations

1. Data should be delivered as service data units.

2. On the sending side, send the specified SpacePacket upon receiving a SpacePacket transmission request from a higher-level application.

   Ex: Devices receiving a TC packet from a satellite control device should execute that TC Packet.
   Ex: Mass memory receiving a TM packet from a sensor device writes the TM packet to the recording area.

10.2.1.1 Packet service communication procedure

Figure 10.1 shows the packet service communication procedure in SpaceWire-R.

10.2.2 Packet service parameters

SpaceWire-R design parameters

The design parameters provided in Applicable Document 5.

Data generation amounts and destinations

Packet length, frequency, and destination of service data units to be generated or transmitted at each device.

Communication channel OPEN/CLOSE timing

The timing for OPEN/CLOSE of each communication channel.

Ex: After turning on the power supply, the state machine of the Transmit TEP is automatically changed from the closed to the enabled state, and the OPEN command is constantly sent.
Ex: SpaceWire-RMAP or SpaceWire-PTP packet services (or memory access services) control higher-level applications managing transmit TEP and transition the transmit TEP state machine from the closed to the enabled state.

Transmit TEP/Receive TEP parameter-setting methods

Method for setting the SpaceWire address or channel number of the receive TEP for each transmit TEP.
Method for setting transmit TEP information (channel number, SpaceWire logical address, etc.) that accepts an OPEN command in each receive TEP.

Service data unit maximum length

Maximum length of service data units transmitted by each device.

Service data unit transmit timing

(When adopting TimeCode-based time division control (§4)) Time slots of each device-destination pair for sending packets.

10.3 Synchronization service time information delivery

In synchronization services using SpaceWire-R, time information that cannot be synchronized with SpaceWire TimeCode is provided from a time master device to each device.

10.3.1 Time delivery stipulations

1. Time information must be in CUC format.

2. Transmitted CUC information should be time information for when the next SpaceWire TimeCode is 0.

3. Nodes receiving CUC information should use the received time information to update internal time information when the SpaceWire TimeCode is 0 or at an internally defined time.

   Note In SpaceWire-R, resends are possible. CUC transmission should be designed with margins for the maximum number of retransmissions at the timing where the TimeCode is 0.

4. SpaceWire specifications make simultaneous delivery to multiple devices impossible. Time master devices should generate an independent SpaceWire packet for each device to be time-delivered, and send it by SpaceWire-R.

5. When using synchronization services, the SpaceWire-R segment size must be at least that of one CUC segment.

10.3.1.1 Time delivery communication procedure

Communication procedures for time distribution using SpaceWire-R are the same as those in Fig. 10.1. On the receiving side, update processing of time information is added after receiving CUC data.

10.3.2 Time distribution parameter

SpaceWire-R designer parameters

The design parameter group provided in Applicable Document 5.

Communications channel OPEN/CLOSE timing

Timing for OPEN/CLOSE of each communication channel.

Ex: After turning on the power supply, the state machine of the Transmit TEP is automatically changed from the closed to the enabled state, and the OPEN command is constantly sent.

Ex: SpaceWire-RMAP or SpaceWire-PTP packet services (or memory access services) control higher-level applications managing transmit TEP and transition the transmit TEP state machine from the closed to the enabled state.

Transmit TEP/Receive TEP parameter-setting methods

Method for setting the SpaceWire address or channel number of the receive TEP for each transmit TEP. Method for setting transmit TEP information (channel number, SpaceWire logical address, etc.) that accepts an OPEN command in each receive TEP.

Target devices for sending time information

Those devices connected to a SpaceWire network that will receive time information.

Timing of time information transmission to each device

(When adopting TimeCode-based time division control ((§4)) Period and phase (time slot) to transmit CUC data to each target device.

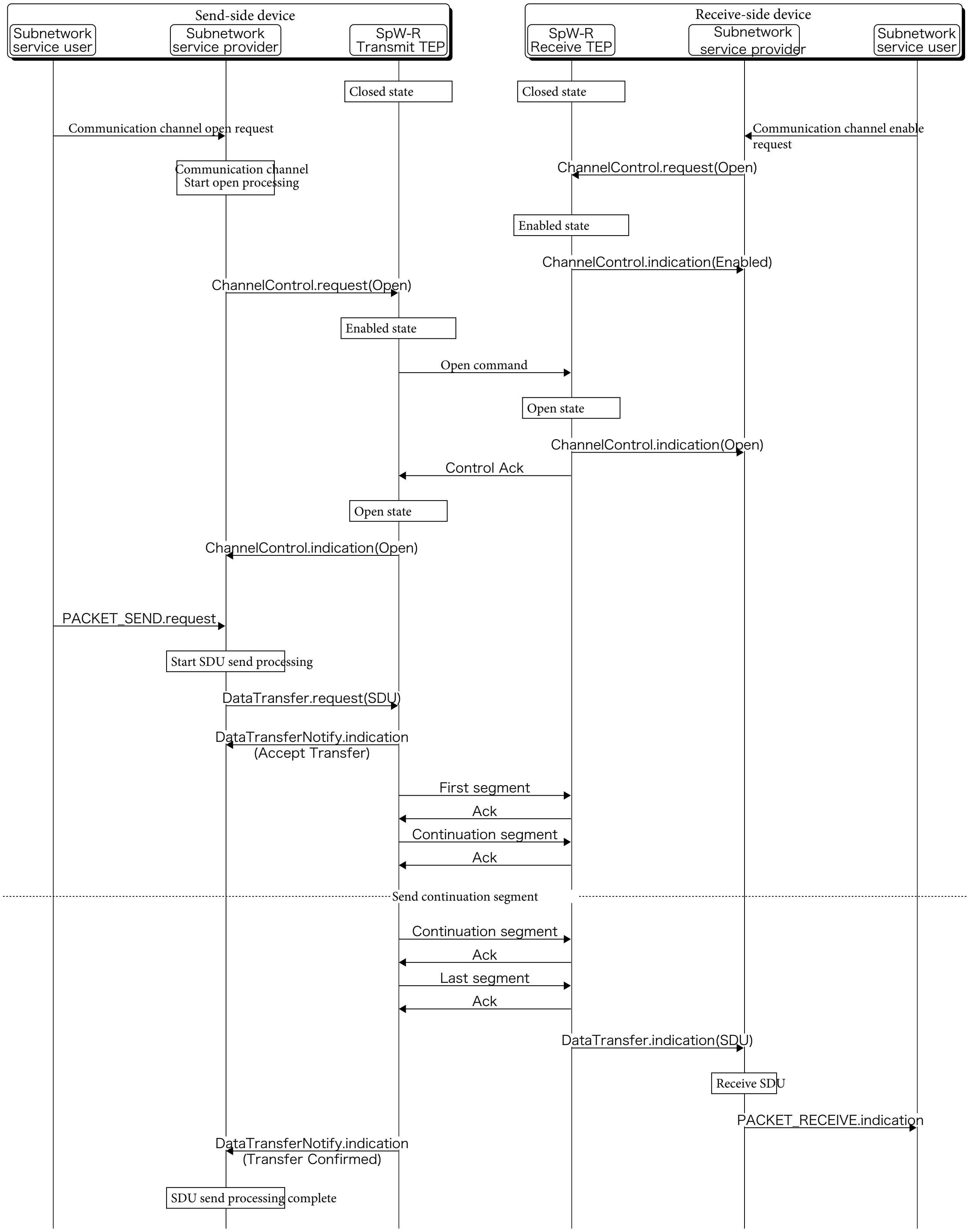

Fig 10.1: Packet service communication procedure in SpaceWire-R