University of Dundee School of Computing

Space
Technology
Centre
University of Dundee

SpaceWire-D

Deterministic Control and Data Delivery Over SpaceWire Networks

Revision: Draft B
Date: April 2010
ESA Contract Number 220774-07-NL/LvH
Ref: SpW-D Draft B

Space Technology Centre
School of Computing
University of Dundee
Dundee, DD1 4HN
Scotland, UK
spacetech.computing.dundee.ac.uk

Document Authors

Steve Parkes

Albert Ferrer Florit

Document Change Log

Date	Revision No	Comments
March 2010	Draft A	Initial draft version
April 2010	Draft B	Advanced Schedule renamed to Multi-Slot Schedule. Example of read operation added to section 4.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 AIMS AND OBJECTIVES

The aim of this document is to describe a method for passing information over a SpaceWire network deterministically i.e. where the time of delivery can be determined a priori within certain bounds. The need for deterministic delivery of information arises from AOCS and GNC systems on board a spacecraft where the time that information is read from a sensor or sent to an actuator is important. In order for SpaceWire networks to be used for both payload and command and control applications a means of providing this determinism is essential. Combining payload and command and control onboard communications networks could significantly reduce system mass and complexity. The solution presented in this document to provide deterministic data delivery does so over SpaceWire networks which fully conform to the ECSS-E-ST-50-12C standard [AD1] i.e. no changes to SpaceWire interfaces or routers are necessary. Furthermore it uses the RMAP protocol, ECSS-E-ST-50-52C [AD3], for transferring information over the SpaceWire network.

The objective of this report is to:

- Describe a protocol where deterministic delivery of information can be provided over an arbitrary SpaceWire network

1.2 GUIDE TO DOCUMENT

Section 2 defines various terms used in this document.

Section 3 describes a method for providing deterministic delivery over an arbitrary SpaceWire network.

Section 4 considers the performance attainable with SpaceWire-D using a simple schedule.

Section 5 outlines the SpW-D protocol stack.

1.3 ACRONYMS AND ABBREVIATIONS

AD	Applicable Document
AOCS	Attitude and Orbit Control System
ECSS	European Cooperation for Space Standardization
GNC	Guidance and Navigation Control
QoS	Quality of Service
RMAP	Remote Memory Access Protocol
RMW	Read/Modify/Write
SpW	SpaceWire

1.4 APPLICABLE DOCUMENTS

The documents applicable to this specification are listed in Table 1-1.

Table 1-1: Applicable Documents

REF	Document Number	Document Title
AD1	ECSS-E-ST-50-12C Formerly ECSS-E50-12A, January 2003	SpaceWire: Links, nodes, routers and networks
AD2	ECSS-E-ST-50-51C	SpaceWire Protocol Identification
AD3	ECSS-E-ST-50-52C	SpaceWire Remote Memory Access Protocol

2 DEFINITIONS

2.1 TERMS DEFINED IN OTHER STANDARDS

For the purpose of this specification, the terms and definitions from the following standards shall apply:

ECSS-S-ST-00-01

ECSS-E-ST-50-51C SpaceWire Protocol Identification

2.2 TERMS SPECIFIC TO THE PRESENT SPECIFICATION

epoch repeat cycle time for time-slot identifiers after which the time-slot numbers repeat

QoS Quality of Service.

Quality of Service level of service that is requested and provided

reliable data gets to the destination if it is at all possible even if there are transitory errors or even permanent errors provided that the network has some appropriate redundancy

schedule order of transactions taking place on the network

schedule table table that specifies when a initiator can send an RMAP command so that together the complete set of schedule tables for all initiators fully defines the order of all transactions taking place on the network

time-slot the interval between two time-codes during which a single RMAP transaction can take place from an initiator

time-slot number incrementing value assigned to successive time-slots for identification purposes, this value is given by the time-value of the time-code that starts a time-slot

time-value the value of a time-code

3 DETERMINISM THROUGH TIME DIVISION MULTIPLEXING

3.1 SPACEWIRE

a) SpaceWire-D shall operate over a network conforming to the SpaceWire standard [AD1].

NOTE: SpaceWire-D does not require any modifications to the SpaceWire standard. It is able to operate using existing SpaceWire interface and router devices.

3.2 RMAP TRANSACTIONS

a) Information shall be passed between SpaceWire nodes using the RMAP protocol [AD3].

NOTE: This permits the reading and writing of registers and memory in a remote target node over a SpaceWire network. These operations are considered sufficient for payload data-handling and command and control applications.

NOTE: SpaceWire-D does not require any modifications to the RMAP standard. There are some constraints on the RMAP implementation e.g. on time taken to respond to an RMAP command, and on the maximum amount of information that can be read or written by a single RMAP command (see section 3.8).

3.3 TIME-SLOTS

a) Time-slots shall be delimited by SpaceWire time-codes.

b) The receipt of a time-code at a node shall indicate the start of a time-slot.

c) The time-slot number shall be the same as the time-value of the time-code that indicates the start of a time-slot.

d) The end of a time-slot shall normally be indicated by the arrival of the next time-code.

NOTE: The relationship between time-codes and time-slots is illustrated in Figure 3-1.

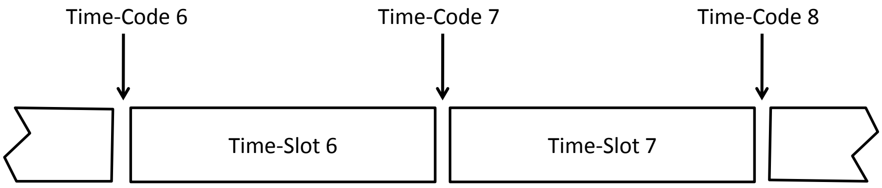

Figure 3-1 Time-Slots

3.4 TIME-CODE WATCHDOG

a) A time-code watchdog timer should be kept in each initiator to check for the correct arrival of each time-code.

b) The time-code watchdog timer shall check for arrival of a time-code sooner than the minimum expected interval between time-codes (early time-code).

c) The time-code watchdog timer shall check for a time-code that does not arrive before the maximum expected interval between time-codes (late time-code).

d) In the event of an early or late time-code the initiator shall flag an error to the user application.

NOTE: The user application could be a local network management application. It is not the purpose of the current specification to define the network management operation.

NOTE: The time-code watchdog is illustrated in Figure 3-2.

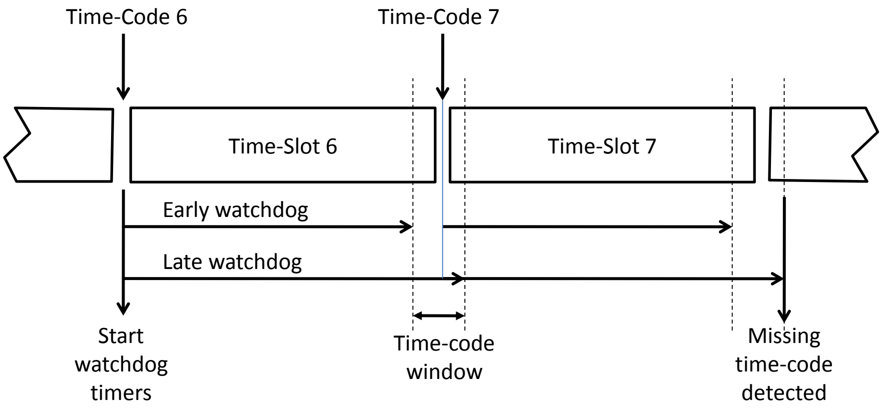

Figure 3-2 Time-Code Watchdog

3.5 SCHEDULE TABLE

a) A schedule shall be provided to ensure deterministic delivery of information.

3.5.1 Simple Schedule

NOTE: The simple schedule gives an initiator full control of the network for one or more specified time-slots. This means that when that initiator is permitted to send an RMAP transaction it may do so to ANY target node on the network. The RMAP transaction must start and finish in the same time-slot.

a) Any RMAP transaction shall start and finish in the same time-slot.

b) Only one RMAP transaction shall take place in a particular time-slot.

c) An initiator shall be permitted to initiate a transaction with any target device during a time-slot in which it is scheduled to initiate RMAP transactions.

d) A simple schedule shall be defined using a schedule table for each initiator which specifies in which time-slot that initiator is allowed to initiate RMAP commands.

EXAMPLE: An example simple schedule table is illustrated in Figure 3-3.

Time-slot	0	1	2	3	...	63
Initiator allowed to initiate an RMAP transaction	Yes	No	No	Yes		No

Figure 3-3 Example Simple Schedule Table

3.5.2 Concurrent Schedule

NOTE: The concurrent schedule makes more efficient use of network bandwidth by allowing more than one initiator to initiate RMAP transactions in a time-slot. This gives rise to the possibility that two initiators might attempt to use the same network resources (SpaceWire links) at the same time. The schedule table has to be constructed to prevent this. This constrains the target nodes that one initiator can send RMAP commands to, while another initiator is also sending commands. Additional network performance is gained at the expense of a more complex schedule.

a) More than one initiator may initiate RMAP transactions in the same time-slot provided that the paths from each of the initiators to their targets do not use any of the same SpaceWire links in the network.

EXAMPLE: A typical application is on board data handling, where a mass memory unit is reading data from each instrument and writing data to a telemetry system, while a control processor is controlling instruments and monitoring housekeeping information. An example schedule table is illustrated in Figure 3-4.

Time-slot	0	1	2	3	...	63
Control Processor Targets	41, 43, 44, 45,	42, 43	42, 43	40, 41, 43, 44,		40
Mass Memory Targets	40	41	41	42		49

Figure 3-4 Example Concurrent Schedule Table

In this example the control processor can initiate RMAP transactions with one of several target devices listed in the schedule table for each time-slot. The mass memory device is gathering information from one target device in each time-slot. The targets that the control process can communicate with are carefully chosen to avoid any network resource conflicts with the transactions that the mass memory device is initiating in each time-slot.

3.5.3 Multi-Slot Schedule

NOTE: The multi-slot schedule builds on the concurrent schedule to improve network efficiency further. Where a large amount of data has to be transferred between two nodes, the RMAP transaction to accomplish this is permitted to occupy more than one adjacent time-slot. This allows more data to be transferred in the one RMAP transaction. The schedule has to ensure that no conflict of network resources occurs over the duration of this extended RMAP transaction. Additional network performance is one again achieved at the expense of a more complex schedule.

a) An RMAP transaction may initiate an RMAP transaction which has a duration of more than one time-slot i.e. the amount of data being written or read exceeds

b) The schedule table shall ensure that when an RMAP transaction has a duration of more than one time-slot, it does not use the same network resources (SpaceWire links) as any other transactions occurring during any of those time-slots.

EXAMPLE: An example schedule table is illustrated in Figure 3-4.

Time-slot	0	1	2	3	...	63
Control Processor Targets	41	42	43			40
Mass Memory Targets	40	41		42		49

Figure 3-5 Example Multi-Slot Schedule Table

In this example the mass memory initiates a long RMAP transaction with target 41 in time-slot 1. This transaction is expected to complete within two time-slots.

3.6 INITIATOR

a) Each node that is capable of being an initiator shall hold a copy of the schedule table.

b) On receipt of a time-code an initiator shall check the schedule to determine if it is allowed to initiate an RMAP command during that time-slot.

c) If the initiator is not allowed to initiate an RMAP command during the current time-slot, it shall not initiate any RMAP commands.

d) If the initiator is allowed to send an RMAP command during the current time-slot, and if it has an RMAP command to send to any target device permitted by the schedule table, it shall send out that RMAP command.

e) RMAP write command shall always request an acknowledgement.

f) RMAP write commands can use verify before write for critical commands.

g) After sending out an RMAP command the initiator shall listen for the reply to the RMAP command.

h) On receipt of the reply to the RMAP command the information that it contains including the status information and any data returned in response to a read command, shall be passed to the user application that initiated the command.

i) If simple or concurrent scheduling is being used and no reply is received by the time the next time-code is received, the initiator shall flag an error to the user application.

j) If advanced scheduling is being used, an initiator may send an RMAP transaction that will take longer than one time-slot interval to complete, as specified by the schedule table.

k) When advanced scheduling is being used and an RMAP transaction longer that one time-slot is being initiated, the initiator shall check for completion of the RMAP transaction at the end of the last time-slot allocated for that transaction and flag an error to the user application if the RMAP transaction has not completed in time.

3.7 TARGET

a) On receipt of an RMAP command the target device shall process it in accordance with the SpaceWire RMAP standard [AD3].

b) Target only nodes shall not need to hold a copy of the schedule table.

c) Target only nodes may hold a copy of the schedule table for fault detection purposes.

3.8 IMPLEMENTATION CONSTRAINTS

3.8.1 Initiator constraints

a) The maximum amount of data that can be read in an RMAP read command or written in an RMAP write command shall be 256 bytes (TBC) when simple or concurrent scheduling is being used.

b) The maximum amount of data may be longer than 256 bytes when advanced scheduling is being used.

c) The time taken from the receipt of a time-code to the starting to send out an RMAP command from an initiator shall be less than 5 µs (TBC).

NOTE: If this is difficult to achieve with a specific implementation of an initiator, operating a local clock synchronised to time-codes might help with achieving this requirement.

3.8.2 Target constraints

a) The time taken from receipt of the complete RMAP command header in a target node to the authorisation or rejection of that RMAP command shall be less than 5 µs (TBC).

b) The latency in transferring data from SpaceWire interface to memory shall be less than 5 µs (TBC).

c) The time taken from completion of writing data to memory to starting to send the RMAP command shall be less than 5 µs (TBC).

4 PERFORMANCE

In this section the anticipated performance of SpaceWire-D is considered.

The operation of an initiator and target performing a write operation during a time-slot is illustrated in Figure 4-1.

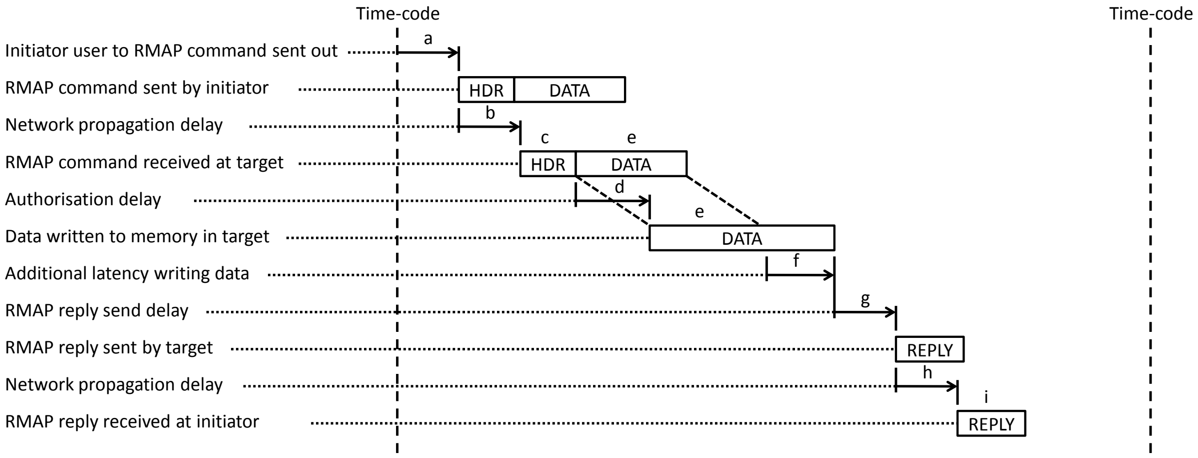

Figure 4-1 Performance of RMAP Write

The receipt of a time-code starts off the activities indicated in Figure 4-1.

a) Interval from receipt of time-code to the RMAP command starting to be sent by the initiator. This interval includes: the time to receive, decode and respond to the time-code; the time to check the schedule table; the time to start to send out the RMAP command (assuming that the command has already been prepared ready for sending). This interval is entirely dependent upon the initiator implementation (see clause 3.8.1(c)).

b) Interval for the SpaceWire packet containing the RMAP command to propagate across the SpaceWire network from initiator to target. This will mainly depend upon the number of routers between the initiator and the furthest target node. Assuming a time delay per router of 0.6 μs, the total propagation delay will be 0.6R where R is the number of routers in the longest path used between an initiator and a target.

c) Interval for sending the RMAP header, including any path address bytes. The size of the RMAP header including the SpW Target Address and Reply Address is H=R+16+P bytes, where R is the number of routers in the path from initiator to target and P is a the closest multiple of 4 which is greater than or equal to R. Thus if there are four router R=4 and P=4, so H=24 bytes. The time to send this header depends upon the SpaceWire data rate, S Mbits/s, and is 10H/S μs. For example with H=24 and S = 200 Mbits/s, T_c = 1.2 μs.

d) Interval for authorising the RMAP command once the header has been received. This is dependent upon the target implementation (see clause 3.8.2(a))).

e) Interval to send the data and data CRC. This is given by 10(D+1)/S, where D is the number of data bytes. For D = 256 (the maximum amount of data permitted in a SpW-D RMAP write command or read reply, see clause 3.8.1(a)), the time to send the data and data CRC is T_e = 12.85 µs when S = 200 Mbits/s.

f) Assuming that the target is able to write data to memory as fast as the SpaceWire network can deliver it, this interval covers any additional latency in the transfer of data from the SpaceWire interface to memory. It is dependent upon the implementation of the target node (see clause 3.8.2(b)).

g) Interval from the completion of writing data to memory in the target to starting to send the RMAP reply. This interval is dependent upon the implementation of the target node (see clause 3.8.2(c)).

h) Interval for the SpaceWire packet containing the RMAP reply to propagate across the SpaceWire network from target to initiator. This will mainly depend upon the number of routers between the initiator and the furthest target node. Assuming a time delay per router of 0.6 µs, the total propagation delay will be 0.6R where R is the number of routers in the longest path used between an initiator and a target. This interval is the same as (b).

i) Interval for sending the RMAP reply, including any path address bytes. The size of the RMAP reply including the Reply Address is E=R+8 bytes, where R is the number of routers in the path from target to initiator. Thus if there are four router R=4, E=12 bytes. The time to send this header depends upon the SpaceWire data rate, S Mbits/s, and is 10E/S µs. For example with E=12 and S = 200 Mbits/s, T_i = 0.6 µs.

The total time for the complete transaction is thus:

$$TT = Ta + Tb + Tc + Td + Te + Tf + Tg + Th + Ti$$

The corresponding performance of a SpaceWire-D network using simple scheduling is illustrated in Figure 4-2.

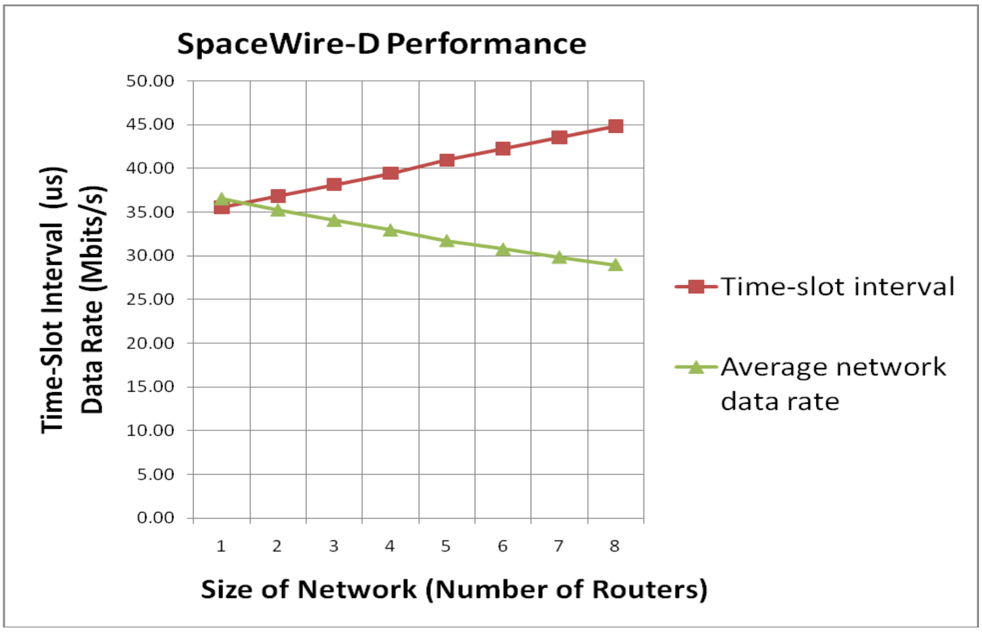

Figure 4-2 Data Rate of Simple Schedule Network vs Size of Network

For this analysis the following assumptions were made:

- The link data rate was 200 Mbits/s,
- A simple schedule was used so only one initiator was initiating transactions at any time, and
- The traffic comprised both payload data transfers (256 bytes per RMAP transaction) and command and control information (4 bytes per RMAP transaction) with an average of 132 bytes per RMAP transaction.

For a network with several layers of routing and with links running at 200 Mbits/s, the overall data rate of the simple schedule system is around 30 Mbits/s (real data rate). Using concurrent or advanced scheduling dramatically increases the aggregate data rate.

Figure 4-3 and Figure 4-4 show the effect of the data field size on the time-slot interval, average data rate and epoch interval.

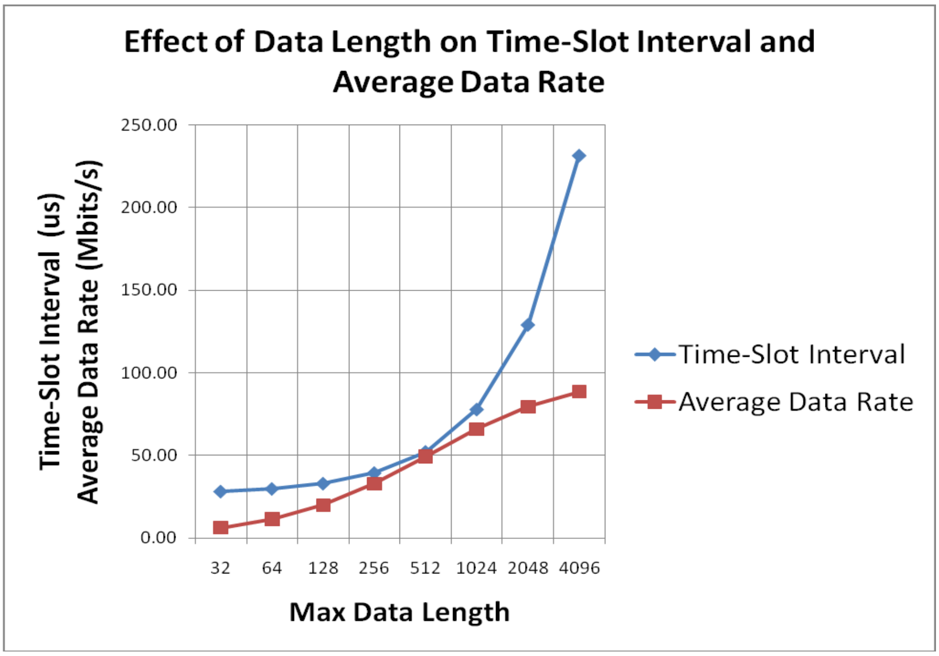

Figure 4-3 Effect of Data Length on Time-Slot Interval and Average Data Rate

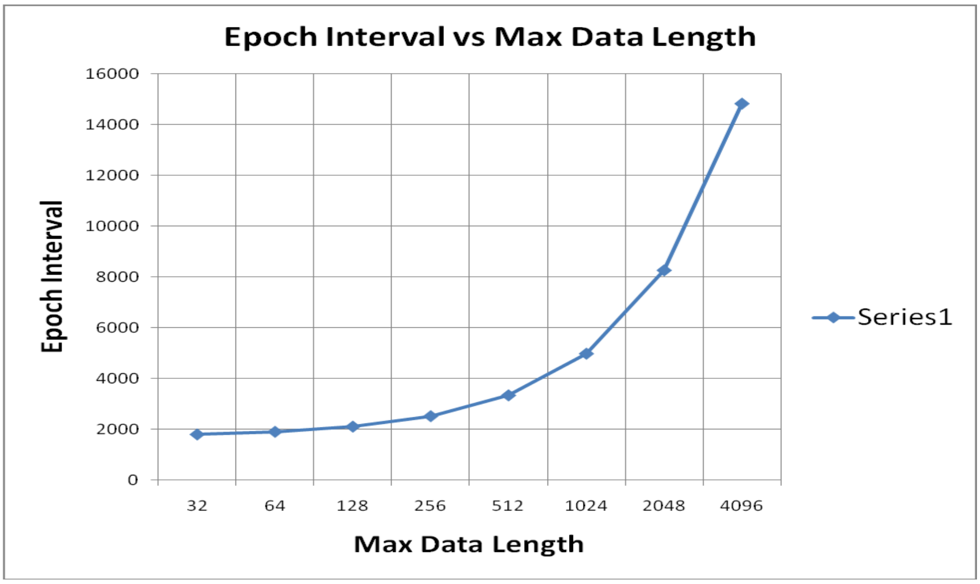

Figure 4-4 Epoch Interval vs Maximum Data Length

The RMAP read operation has similar timing characteristics as illustrated in Figure 4-5.

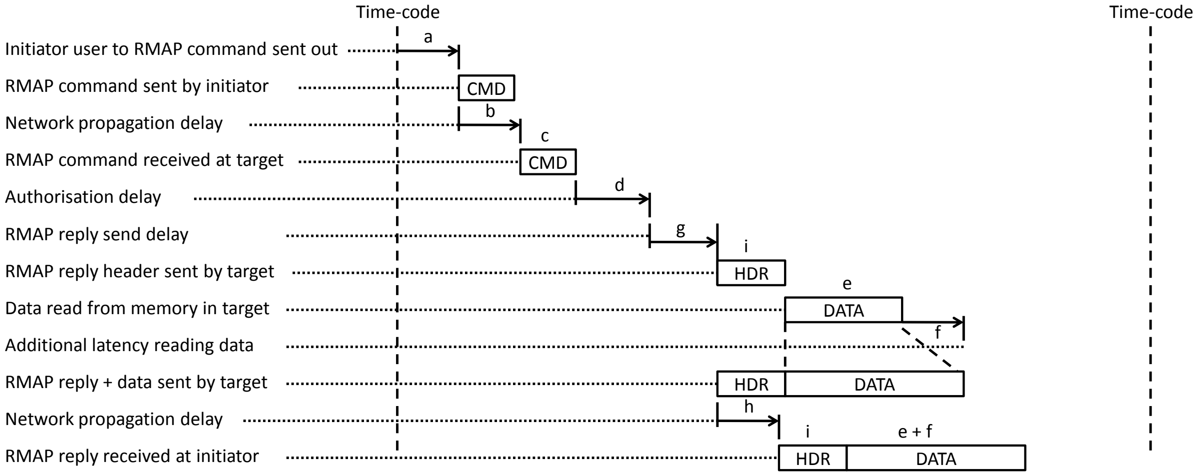

Figure 4-5 Performance of RMAP Read

The delays for the read command are very similar to those of the write command. The letters used to denote each of the time delays for the read command are the same as the comparable delays in the write command (hence they are not in alphabetical order in Figure 4-5).

The receipt of a time-code starts off the activities indicated in Figure 4-5.

a) Interval from receipt of time-code to RMAP read command starting to be sent by initiator.

b) Interval for the SpaceWire packet containing the RMAP read command to propagate across the SpaceWire network from initiator to target.

c) Interval for sending the RMAP read command, including any path address bytes.

d) Interval for authorising the RMAP read command once it has been received.

g) Interval from authorisation of the read command to starting to send the RMAP reply.

h) Interval for SpaceWire packet containing the RMAP reply to propagate across the SpaceWire network from target to initiator.

i) Interval for sending the RMAP reply header, including any path address bytes.

e) Interval to send the data and data CRC.

f) Interval to cover any additional latency in the transfer of data from the memory to the SpaceWire interface.

The total time for the complete transaction is thus:

$$TT = Ta + Tb + Tc + Td + Tg + Th + Ti + Te + Tf$$

which is the same as for the write command. Hence the performance is the same for either read or writes operations.

5 SPACEWIRE-D PROTOCOL STACK

In this section the SpaceWire-D protocol stack is outlined along with the services that it provides.

5.1 DETERMINISTIC SPACEWIRE PROTOCOL STACK

The protocol stack for SpaceWire-D is illustrated in Figure 5-1.

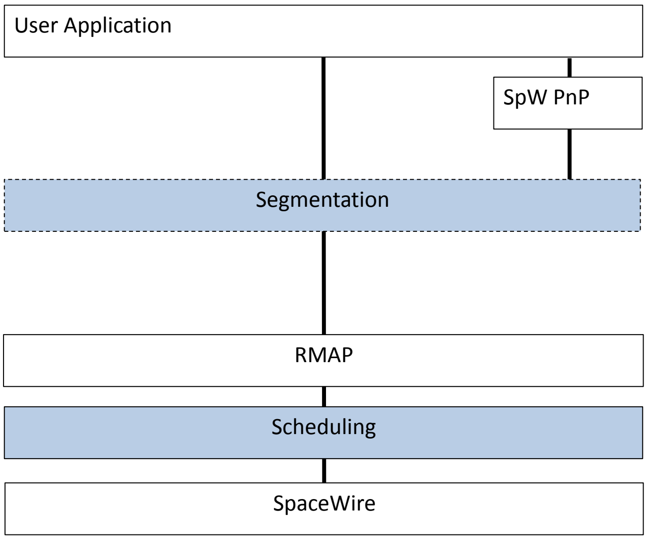

Figure 5-1 SpaceWire-D Protocol Stack

SpaceWire-D provides medium access control using a schedule which specifies when a particular initiator is allowed to initiate an RMAP transaction. The scheduling layer which is the main part of SpaceWire-D, sits between RMAP and SpaceWire.

The other function that is required by SpaceWire-D is segmentation, since the maximum length of an RMAP data field is limited to 256 (TBC) bytes (unless advanced scheduling is being used). SpaceWire-D passes the responsibility for segmentation to the user application by restricting the size of the data field in any RMAP transaction it is requested to initiate.

Apart from this restriction in the size of the data field, SpaceWire-D is able to perform any required RMAP transaction. Since the proposed Plug-and-Play (PnP) protocol for SpaceWire uses RMAP it means that PnP will also operate over a SpaceWire-D network, provided that the size of the data fields used in PnP do not exceed 256 bytes (TBC).

If reliability mechanisms (retry and redundancy) are required to be provided for SpaceWire-D then they could make use of the acknowledgements provided by RMAP and initiate retry and redundancy switching if no acknowledgement is received by the end of the time-slot. This layer is illustrated in Figure 5-2.

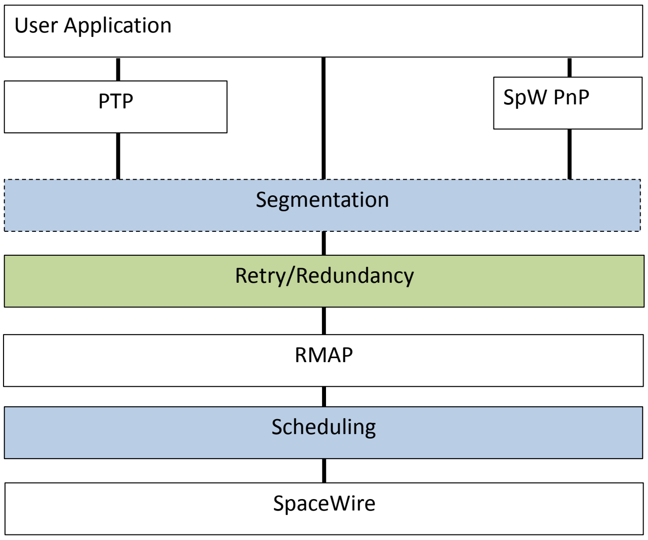

Figure 5-2 SpaceWire-D Protocol Stack with Retry/Redundancy

A further possibility is illustrated in Figure 5-2, which is a packet transfer protocol (PTP). Such a protocol could be operated using RMAP to write to a packet buffer in a target node. Both a push and a pull type of packet transfer capability could be provided using RMAP writes or reads respectively.

5.2 DETERMINISTIC SPACEWIRE SERVICES

SpaceWire-D provides the following services:

- Write to remote memory
- Read from remote memory
- Read-modify-write remote memory

The service interfaces are identical to those for RMAP [AD3].

SpaceWire-D could be expanded to include retry/redundancy, PTP support, PnP support and possibly explicit segmentation.