DOCUMENT TYPE: Proposal of a SpaceWire networking Protocol

TITLE: MULTIPLE ACCESS COMMUNICATION PROTOCOL

DOCUMENT No.: TL xxxxx **PAGE:** I of V, 23

PROJECT Ref.:

ISSUE No.: Draft A

PREPARED BY: G. Saldi

CHECKED BY: M. De Meo

PROJECT LEADER: **DATE:**

PAPM: **DATE:**

PROGRAM MANAGER: **DATE:**

CONFIGURATION:

DATE FOR APPROVAL:

DISTRIBUTION LIST

POS.	NAME	DEPT.	N°
	M. De Meo	LABEN	1
	G. Saldi	LABEN	1
	A. Marra	LABEN	1
	M.L. Esposti	LABEN	1

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release
Draft A	JAN. 05	ALL	FIRST DRAFT OF THE DOCUMENT	---

TABLE OF CONTENTS

LIST OF FIGURES

1. ABSTRACT

AD-1 establishes an efficient way to interconnect processing nodes and I/O nodes through a SpaceWire physical network, based on cables and routers; in particular AD-1 defines the exchange of simplified packets consisting of a routing label, a cargo and an end of packet marker.

Since the communication will finally occur, rather than among nodes, among the tasks spread over the physical nodes, a more structured SpaceWire networking Protocol (SnP), complying the protocol identification normative (AD-2), should be devised.

Let' s image each configuration of a spacecraft mission like a set of tasks simultaneously and concurrently running on the various nodes of a SpaceWire network. At any time each task needs new inputs and generates new outputs; in general:

- the inputs of a specific task come from the outputs of one or more other tasks;
- the outputs of a specific task go to the inputs of one or more other tasks.

This viewpoint leads to the identification of a virtual network, consisting of virtual paths, dedicated to the exchange of the I/O data among the various tasks relevant to a specific mission configuration. Each task will have corresponding I/O mailboxes to exchange its I/O data with other tasks through the virtual network.

2. KEYWORDS

SpaceWire
Virtual Network
Virtual Path
Virtual Channel
Mailbox
.

3. INTRODUCTION

This document proposes a first draft of a SpaceWire networking Protocol named Multiple Access Communication Protocol. All the content is therefore matter of discussion and comments will be welcome (saldi.g@laben.it).

3.1. PURPOSE AND SCOPE

Purpose of this protocol is to plan the communication among the various tasks running at any given time inside the various nodes of a SpaceWire network, by combining together:

- security in the configuration, commanding and monitoring performed by a supervisor node on all the nodes of the SpaceWire network;
- high data throughput in the transfer of scientific data among the SpaceWire nodes interfacing instruments, on board mass memories, telemetry, ...etc.

3.2. STRUCTURE OF THE DOCUMENT

Sections 5.1 to 5.3 provide an overview of the protocol operations; in particular section 5.3 highlights correspondences and differences with RD-1.

Sections 5.5 to 5.16, still to be completed, introduce the protocol details here intentionally limited to the format of the packets exchanged during each transaction. This level of detail should be however sufficient to orient the reader (particularly if member of the SpaceWire Working Group) towards a good understanding of the protocol operations, which will be exhaustively dealt with a later issue of this document.

Section 5.17 lists open points waiting for assessment.

4. APPLICABLE AND REFERENCE DOCUMENTS

4.1. APPLICABLE DOCUMENTS

AD-1 ECSS-E-50-12A – SpaceWire - Links, nodes, routers and networks - 24 January 2003 (available at http://www.estec.esa.nl/tech/spacewire/).

AD-2 ECSS-E-50-12B Draft A – Chapter 5 – Protocol identification (normative) – Nov. 2004 (available at http://www.estec.esa.nl/conferences/01C25)

4.2. REFERENCE DOCUMENTS

RD-1 ECSS-E-50-12B Draft A – Chapter 6 – Remote Memory Access Protocol (normative) – Nov. 2004 (available at http://www.estec.esa.nl/conferences/01C25)

5. MULTIPLE ACCESS COMMUNICATION PROTOCOL

5.1. GENERAL

The virtual network interconnecting the tasks simultaneously and concurrently running on the various nodes of the SpaceWire network, is formed by many concurrent virtual paths, which time-share the physical paths of the network. A virtual path is a point to point connection, between two remote tasks, based on:

- a TX mailbox in the source node
- a physical path through the SpaceWire network (e.g. identified by the logical address of the destination node)
- a RX mailbox in the destination node.

Any (set of) I/O data exchanged on a virtual path as a sequence of SpaceWire packets, will have inside the header of each packet, after the mandatory Protocol Identifier (see AD-2), the identifier of the mailbox in the destination node.

Inside a node a Virtual Channel is a full duplex I/O channel, used to exchange two independent incoming and outgoing data flows; it is supported by a RX/TX mailbox pair which behaves as terminal of one input and one output virtual path.

A new mission configuration, necessary after a node failure or to run a new set of application tasks spread over the various nodes, will require to re-program the virtual network. If the identifiers of the I/O mailboxes of any virtual path shall depend on the corresponding task and not on the node running the task, only physical path re-programming will be necessary and limited to:

- the routing tables of the SpaceWire routers (227 bytes per router);
- the (optional) Destination Address bytes and the Logical Address byte, on top of each packet later on sent by any given task, identifying the new physical path (i.e. the destination node) associated to each virtual path.

By complying this protocol, a SpaceWire node will access the memory space of the other nodes by addressing mailboxes rather than specific memory cells (indirect versus direct addressing). Being the mailbox size pre-defined in a receiving node, the risk that a packet longer than the expected one overflows the destination memory, is prevented. However the protocol allows also elementary read/write transactions based on direct addressing.

The protocol foresees:

- a set of bi-directional transactions between a master and a slave node having purpose of command, configuration and monitoring.
- one unidirectional transaction between a couple of nodes having purpose of high throughput and low latency data exchanges.
- one unidirectional transaction between a slave and a master node having purpose of interrupt notification.

Though most of the protocol complexity is concerned with command, configuration and monitoring (bi-directional) transactions, this protocol plans to allot most of the SpaceWire network bandwidth to (unidirectional) transactions providing exchanges of scientific data.

The following packet types are foreseen:

- Command packet
- Reply packet
- Data packet
- Interrupt message packet.

5.2. TYPE OF TRANSACTIONS

The following transactions are foreseen:

- Bi-directional short read transaction
- Bi-directional short write transaction (2 cases)
- Bi-directional virtual channel initialisation transaction
- Bi-directional virtual channel monitoring transaction
- Bi-directional virtual channel/file load transaction (3 cases)
- Bi-directional mem. to I/O (I/O to mem.) Transaction (4 cases)
- Bi-directional file move transaction (4 cases)
- Bi-directional long read transaction (2 cases)
- Bi-directional long write transaction (4 cases)
- Unidirectional data exchange transaction
- Unidirectional interrupt transaction

The bi-directional transactions are summarised in Figure 5.5-1, showing that other 10 types of bi-directional transactions are reserved for future expansion (t.b.d. options).

Each bi-directional transaction is based on the exchange of a forward command packet and of a backward reply packet between a master node and a slave node.
An exception to this rule is the file load transaction, where the master node sends the command packet, receives the packet sequence carrying the file, and then the reply packet (which is omitted in 1 of 3 cases – see Figure 5.5-1).

Bi-directional short/long read/write transactions access the destination node via direct memory addressing and provide the same service of the corresponding transactions of RD-1. All the other bi-directional and unidirectional transactions are based on indirect memory addressing.

All the bi-directional transactions, but long read/write transactions, are oriented to the command, control and configuration of slave nodes. For these transactions the command will be executed inside the slave node only after validation of the complete command packet (i.e. the packet is discarded when the header checksum or the data checksum fails). Therefore, if data checksum fails, wrong data have not been written into the destination memory (reserved to sensitive configuration data). The data length of the exchanged packets is limited to fix the depth of the buffer needed for data checksum validation inside the receiving node. These transactions can be used to prepare the exchange of large data amount (e.g. data frame sequences) between couples of nodes (e.g. a camera and a SSMM); in particular:

- A bi-directional short read/write transaction provides the access, inside the destination node, to a single memory cell or register containing command or configuration parameters.
- A bi-directional virtual channel initialisation transaction transfers (within a single command packet), from the master node to a slave node, all the parameters needed to configure the RX/TX mailbox pair of a virtual channel, which is contextually enabled to exchange data..
- A bi-directional virtual channel monitoring transaction transfers (in a single reply packet), from a slave node to the master node, all the parameters containing the status of a virtual channel (e.g. after completion of a unidirectional data exchange transaction).
- A bi-directional virtual channel/file load transaction drives the slave node to transmit the content of a mailbox to the master node, or to another node in the SpaceWire network.
- A bi-directional mem. to I/O (or I/O to mem.) transaction (4 ways) drives the slave node to move data between a memory area and a specific I/O port mapped inside its address space. Data can be moved starting from the first or last row or column for easy “corner turning”.
- A bi-directional file move transaction (4 ways) drives the slave node to move data from a old to a new memory area mapped in the slave node address space. Since old and new memory area can overlap, data can be moved starting from the first or last row or column.

A bi-directional long read/write transaction provides the transfer of a large data packet directly between the master and a slave node (see RD-1). For these transactions the command will be executed inside the slave node after validation of the command packet header, but data will be written in the addressed memory before data checksum validation. Therefore, if data checksum fails, wrong data have however been written into the destination memory.

An unidirectional data exchange transaction provides the exchange of a sequence of packets (e.g. a video data frame on line basis) between a source node and a destination node (e.g. a camera and a SSMM), both previously configured (either locally - e.g. master node -, or remotely - e.g. slave node) to exec this data exchange.

An unidirectional interrupt transaction provides the transfer of a single packet, from a slave node (configured to be remotely controllable) to the master node, containing an interrupt message. Inside a slave node both anomalous conditions and normal end of operations can be notified, with min. latency, to the master node.

5.3. DEFINITIONS

5.3.4. CORRESPONDECES WITH THE REMOTE MEMORY ACCESS PROTOCOL

Most of the parameters defined in section 6.2 of RD-1 maintain the same meaning and are processed in the same way in both command and reply packets.
In particular this is valid for:

- Destination Address bytes (specifying the physical path to reach the destination node)
- Destination Node Logical Address byte (carrying the logical address of the destination node)
- Protocol Identifier byte (and related extension bytes) as per AD-2
- Extra Return Address words, inside the command packet to specify the return physical path to be used in the reply packet
- Return Node Logical Address byte, inside the command packet to specify the return Logical Address to be used in the reply packet.
- Transaction Identifier
- Header Checksum and Data Checksum.

The parameters mentioned in the first 3 and the last of the above points are relevant also to data packets and interrupt message packets.
Refer to section 6.2 of RD-1 for the detailed definition of the above parameters.

5.3.5. DIFFERENCES WITH THE REMOTE MEMORY ACCESS PROTOCOL

The main differences with respect to RD-1 are concerned with the following parameters:

- The “Virtual Channel Identifier” (2 bytes) is a new parameter introduced to identify a virtual destination inside the receiving node. In fact, inside a node having a specific Logical Address in the SpaceWire network, many different virtual destinations may exist; these can be data mailboxes used to support uni-directional data exchange, or control mailboxes used inside a slave node to support bi-directional command/reply transactions. The “Virtual Channel Identifier” is thus required to reach the destination mailbox inside the receiving node.
  A command packet (from the master node) will carry both the Destination and the Return Virtual Channel Identifier: the former identifying, inside the slave node, the mailbox used to store and exec the command, the latter identifying, inside the Master Node, the mailbox used to accumulate reply packets.
  Reply, Data and Interrupt message packets contain only the “Virtual Channel Identifier” of their destination node.
  (see also section 5.17, open point a).
- The optional “Password” (4 bytes) is a new parameter used to prevent unauthorised or inadvertent access to the destination memory of a virtual channel. Though superfluous in a closed communication system, it can improve the security in case header and data checksums become useless since computed on a wrong packet; e.g. when a SEU changes one bit of a command packet stored in the local memory of the master node, just before its transmission (i.e. before computing checksums). The presence of a Password in a packet (command, reply, data or interrupt message) is notified by a bit of the below introduced “Type/Command/Status/Sizes” field. The node receiving a packet

will perform the check on the password if this is present in the packet and if, inside the same node, the password check on the addressed mailbox has been previously enabled. The packet is discarded in case of invalid password (see also section 5.17, open point f).

- The optional “Line Identifier” (4 bytes) is a new parameter used in data packets only. It has a different value (e.g. a sequential count) for each data packet of the sequence transferred through a unidirectional data exchange transaction. E.g. the transfer of a video data frame from a camera to a SSMM can be performed by transmitting as many data packets as the lines of the video frame. This parameter allows re-ordering of the data lines at the receiver, mandatory when the SpaceWire network provides multiple physical paths between the source and the destination node of the transaction. The presence of the Line Identifier in a data packet is notified by a bit of the below introduced “Type/Command/Status/Sizes” field.
- The “Type/Command/Status/Sizes” (2 bytes) replaces the equivalent field of RD-1 (see Fig. 1 and Fig. 7 of RD-1); it is used to specify packet type, command, return address length, reply status and other features specific of each packet type, as detailed in section 5.5.

Other differences with respect to RD-1 are:

- “Read/Write Address” of RD-1 has become “Destination Bus Address” (4 bytes) for read/write transactions based on direct addressing, or “Destination Mailbox Identifier” for the other bi-directional transactions based on indirect addressing (see also section 5.17, open point e)
- “Extn Destination Bus Address” (1 byte) is a new parameter which, combined with the 32-bit “Destination Bus Address”, allows read/write transactions with 40-bit address, suitable to access slave nodes with wide memory (e.g. SSMM).
- “Data Length” (4 bytes instead of 3) specifies the number of data samples, which may have a size different from a single octet, being the sample size specified in the Type/Command/Status/Sizes field (see also section 5.17, open point d).
- A reply packet, besides the Transaction Identifier, repeats all the parameters of the corresponding command packet (see also section 5.17, open point h)
- The Data Checksum validation for the command packets containing configuration data (see Figure 5.5-1 those with Command Code<=16) shall be performed, by the node receiving the packet, before storing data into the destination memory.
- The command and reply packets contain “pads” to get a 32-bit word alignment, suitable to microprocessors as those used in space applications (ERC32, Leon, ADSP21020).
- The command (and reply) packets of the various bi-directional transactions, share a common format which can be discriminated on the basis of the command code contained in the Type/Command/Status/Sizes field (detailed in section 5.5). All the command packets have the same size (in 32-bit words) which may differ only for the optional parameters (Dest. Address, Extra Return Address and Password); the same apply to reply packets.

5.4. NOTES ON THE LOOK OF THE PACKET FORMATS

The following notes apply to the look of the packet formats introduced in sections 5.5 to 5.16:

- The optional Destination Address bytes (which don't contribute to packet header checksum) are written in red over a gray background.
- The optional bits of the "Type/Command/Status/Sizes" field which could become don't care or reserved, are written in black over a yellow background.
- The bytes of optional parameters, whose presence is notified in the "Type/Command/Status/Sizes" field of the packet, are written in black over a gray background.
- The bytes corresponding to pads, or to parameters which could be replaced by pads, are written in black over a yellow background.

5.5. FORMAT OF THE "TYPE/COMMAND/STATUS/SIZES" FIELD

The "Type/Command/Status/Sizes" field in a Command Packet includes:

- 5-bit command code to select 1 of 32 commands (i.e. bi-directional transactions);
- 1 bit to notify the presence of the Password;
- 2 bits specifying the number of 32-bit Extra Return Address Word (as per RD-1);
- 3 bits to specify the size in octets of each data sample contained in the packet.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type = 00		Command Code					Pass word Pres ent	Data Sample Size (N. of Octets - 1)			x	x	x	Extra Return Address Words	

Command Code	Command Description
0	read bus cycle and REPLY packet
1	write bus cycle and REPLY packet with data read back from the just written location
2	VC/file acquisition (i.e. load a file) towards the RSC without REPLY packet
3	VC/file acquisition (i.e. load a file) towards the RSC and REPLY packet towards the RSC
4	VC/file acquisition (i.e. load a file) towards a PRSD and REPLY packet towards the RSC
5	write bus cycle and REPLY packet with data copied from the COMMAND packet
6	VC initialisation, always with REPLY packet
7	VC monitoring, always with REPLY packet
8	Mem to I/O or I/O to Mem. transfer with REPLY packet (from the first row)
9	Mem to I/O or I/O to Mem. transfer with REPLY packet (from the last row)
10	Mem to I/O or I/O to Mem. transfer with REPLY packet (from the first column)
11	Mem to I/O or I/O to Mem. transfer with REPLY packet (from the last column)
12	file move up, with REPLY packet (Mem. to Mem. data transfer from the first row)
13	file move down, with REPLY packet (Mem. to Mem. data transfer from the last row)
14	file move left, with REPLY packet (Mem. to Mem. data transfer from the first column)
15	file move right, with REPLY packet (Mem. to Mem. data transfer from the last column)
16	read data block with address increment and REPLY packet
17	write data block with address increment and REPLY packet with data read back
18	write data block with address increment and REPLY packet without any data
19	read data block without address increment and REPLY packet
20	write data block without address increment and REPLY packet with data read back
21	write data block without address increment and REPLY packet without any data
22 to 31	reserved

Figure 5.5-1 - "Type/Command/Status/Sizes" Field in a Command Packet

The “Type/Command/Status/Sizes” field in a Reply Packet has the same content of the corresponding command packet except for:

- the packet type which is changed into 01 for reply packet
- the 3 bit reply status which is added for Ack/No Ack (see also sec. 5.17, open point i)
- the bit notifying the presence of the Password.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type = 01		Command Code					Pass word Pres ent	Data Sample Size (N. of Octets - 1)			Ack/No Ack Ack= 000 No Ack= Non Zero Error Code			Extra Return Address Words	

Figure 5.5-2 - "Type/Command/Status/Sizes" Field in a Reply Packet

The “Type/Command/Status/Sizes” field in a Data Packet includes:

- 3 bits to notify the presence of the Password, Line Identifier and Data Length;
- 6 bits to specify the size in bit of each data sample contained in the packet, where data samples may be not aligned with the data characters to improve bandwidth utilisation. E.g. it is allowed to insert two 12-bit data samples from an A/D converter into 3 data characters on the Spacewire link. (see also section 5.17, open point g)
- The 2 bits specifying the number of Extra Dest. Address Word are used by the transmitter of these packets and not by the receiver. Thus they can be omitted.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type = 10		x	x	x	Data Leng th Pres ent	Line ID Pres ent	Pass word Pres ent	Data Sample Size (N. of bits - 1)						Extra Dest. Address Words	

Figure 5.5-3 - "Type/Command/Status/Sizes" Field in a Data Packet

The “Type/Command/Status/Sizes” field in a Interrupt Message Packet includes:

- 5-bit interrupt code allows to select 1 of 32 interrupt sources
- 1 bit to notify the presence of the Password (see also section 5.17, open point f)
- 3 bits to specify the size in octets of each data sample contained in the packet (see also section 5.17, open point g)
- The 2 bits specifying the number of Extra Dest. Address Word are used by the transmitter of these packets and not by the receiver. Thus they can be omitted.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type = 11		Interrupt Code					Pass word Pres ent	Data Sample Size (N. of Octets - 1)			x	x	x	Extra Dest. Address Words	

Figure 5.5-4 - "Type/Command/Status/Sizes" Field in an Interrupt Message Packet

5.6. BI-DIRECTIONAL SHORT READ TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0
Command Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.6-1 - Bi-Directional Short Read Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1
Reply Header Checksum			
Destination Bus Read Data (1 sample of 1, 2, 3 .... 8 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.6-2 - Bi-Directional Short Read Transaction (Reply Packet)

5.7. BI-DIRECTIONAL SHORT WRITE TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1
Header Checksum			
Destination Bus Write Data (1 sample of 1, 2, 3 .... 8 bytes)			
	Data Checksum	EOP	

Last Byte Transmitted

Figure 5.7-1 - Bi-Directional Short Write Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1	Data Length (N. of samples) =1
Reply Header Checksum			
Destination Bus Written Data (1 sample of 1, 2, 3 .... 8 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.7-2 - Bi-Directional Short Write Transaction (Reply Packet)

5.8. BI-DIRECTIONAL VIRTUAL CHANNEL INITIALISATION TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words
Header Checksum			
Destination Mailbox Config Data (<=32 samples of 4 bytes)			
	Data Checksum	EOP	

Last Byte Transmitted

Figure 5.8-1 - Bi-Directional Virtual Channel Initialisation Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words
Reply Header Checksum			
Destination Mailbox Config Data (<=32 samples of 4 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.8-2 - Bi-Directional Virtual Channel Initialisation Transaction (Reply Packet)

5.9. BI-DIRECTIONAL VIRTUAL CHANNEL MONITORING TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words
Command Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.9-1 - Bi-Directional Virtual Channel Monitoring Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words	Data Length (N. of samples) <= 32x32-bit words
Reply Header Checksum			
Destination Mailbox Config Data(<=32 samples of 4 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.9-2 - Bi-Directional Virtual Channel Monitoring Transaction (Reply Packet)

5.10. BI-DIRECTIONAL VIRTUAL CHANNEL/FILE LOAD TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0
Command Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.10-1 - Bi-Directional Virtual Channel/File Load Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0
Reply Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.10-2 - Bi-Directional Virtual Channel/File Load Transaction (Reply Packet)

5.11. BI-DIRECTIONAL MEM. TO I/O (I/O TO MEM.) TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0
Command Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.11-1 - Bi-Directional Mem. to I/O (I/O to Mem.) Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0	Data Length (N. of samples) =0
Reply Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.11-2 - Bi-Directional Mem. to I/O (I/O to Mem.) Transaction (Reply Packet)

5.12. BI-DIRECTIONAL FILE MOVE TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words
Header Checksum			
Next Position of Addressed File (<=4 samples of 4 bytes)			
	Data Checksum	EOP	

Last Byte Transmitted

Figure 5.12-1 - Bi-Directional File Move Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	PAD
PAD	PAD	Destination Mailbox Identifier	Destination Mailbox Identifier
Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words	Data Length (N. of samples) <= 4x32-bit words
Reply Header Checksum			
Next Position of Addressed File (<=4 samples of 4 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.12-2 - Bi-Directional File Move Transaction (Reply Packet)

5.13. BI-DIRECTIONAL LONG READ TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)
Command Packet Checksum	EOP		

Last Byte Transmitted

Figure 5.13-1 - Bi-Directional Long Read Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)
Reply Header Checksum			
Destination Bus Read Data (N samples of 1, 2, 3 .... 8 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.13-2 - Bi-Directional Long Read Transaction (Reply Packet)

5.14. BI-DIRECTIONAL LONG WRITE TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (= 0÷7)	Dest. Virtual Channel Identifier (= 0÷7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Return Address	Return Address	Return Address	Return Address
Return Node Logical Address	Transaction Identifier	Transaction Identifier	Transaction Identifier
Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)	PAD	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)
Header Checksum			
Destination Bus Write Data (N samples of 1, 2, 3 .... 8 bytes)			
	Data Checksum	EOP	

Figure 5.14-1 - Bi-Directional Long Write Transaction (Command Packet)

First Byte Transmitted

	Return Address	Return Address	Return Address
Return Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Return Virtual Channel Identifier (>7)	Return Virtual Channel Identifier (>7)
Return VC Password	Return VC Password	Return VC Password	Return VC Password
Transaction Identifier	Transaction Identifier	Transaction Identifier	Extn Destination Bus Address
Destination Bus Address	Destination Bus Address	Destination Bus Address	Destination Bus Address
Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)	Data Length (N. of samples)
Reply Header Checksum			
Destination Bus Written Data (N samples of 1, 2, 3 .... 8 bytes)			
	Reply Data Checksum	EOP	

Last Byte Transmitted

Figure 5.14-2 - Bi-Directional Long Write Transaction (Reply Packet)

5.15. UNIDIRECTIONAL DATA EXCHANGE TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (>7)	Dest. Virtual Channel Identifier (>7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Line Identifier	Line Identifier	Line Identifier	Line Identifier
1st Sample = Data Length (opt.)	1st Sample = Data Length (opt.)	1st Sample = Data Length (opt.)	1st Sample = Data Length (opt.)
Header Checksum			
Destination Mailbox Current Line Data Samples (reasonably less than 2^32 Data Samples of up to 64 bits)			
	Data Checksum	EOP	

Last Byte Transmitted

Figure 5.15-1 - Unidirectional Data Exchange Transaction (1 of N Node to Node Data Packet)

5.16. UNIDIRECTIONAL INTERRUPT TRANSACTION

First Byte Transmitted

	Destination Address	Destination Address	Destination Address
Dest. Node Logical Address	Protocol Identifier	Extn Protocol Identifier	Extn Protocol Identifier
Type/Command/Status/Sizes	Type/Command/Status/Sizes	Dest. Virtual Channel Identifier (>7)	Dest. Virtual Channel Identifier (>7)
Destination VC Password	Destination VC Password	Destination VC Password	Destination VC Password
Data Length	Data Length	Data Length	Data Length
Header Checksum			
Interrupt Status Data (Source Node Logical Address, Time Tag Data, ...etc.) (up to M Data Samples of 1, 2, 3 .... 8 bytes)			
	Data Checksum	EOP	

Last Byte Transmitted

Figure 5.16-1 - Unidirectional Interrupt Transaction (Slave Node to Master Node Interrupt Message Packet)

5.17. OPEN POINTS

a. Do we leave a Destination Virtual Channel Identifier of specific value (= 0÷7) in the Command Packet to address few control mailboxes in the destination node? It is redundant with packet type (=10) but it is suitable to save commonality among the formats of the 4 packet types (where this parameter follows the 2 bytes of the “Type/Command/Status/Sizes” field in each packet).
b. Data Length in all command packets and reply packets (i.e. even in those corresponding to the first 16 Command Codes) ? It may be redundant for command codes<16 (configuration commands for which the indicated max length is recommended), but it could be needed for uniformity with command codes >=16 (long read/write commands) and it is suitable to save a common format among command packets and among reply packets (where the “Data Length” bytes are placed just before the packet header checksum).
c. Is the Source Node Logical Address advisable in Data Packets (unidirectional transactions) ? It provides to the receiving node the logical address of the node source of the Data Packet and may be useful for data transfers not supervised by a master node. Currently this parameter doesn’t appear in Figure 5.15-1.
d. Data Length Size: 2, 3 or 4 bytes? 4 is proposed for 32 bit alignment, even if 2 or 3 may be preferable for saving of network bandwidth.
e. Destination Mailbox Address size: 2 bytes should be sufficient, while 4 bytes, as used for the Destination Bus Address, are suitable to save a common format for all the command packets (where these 4 bytes are placed just before the Data Length bytes of each command and reply packet).
f. Since security is more concerned with command transactions rather than with data transactions, the "Destination VC Password" in a Data Packet could be replaced by the "Frame Identifier", used to check if the received Data Packet contains a Line of the currently exchanged Frame. In practice this “Frame Identifier” would become the key to access data mailboxes, while the "Destination VC Password" remains the key to access control mailboxes; what about this possibility?
g. In a Data Packet the parameter " Data Sample Size (=N. of bits - 1) " might be not necessary. In fact the size of each Data Sample is programmed in advance of the exchange of the data packets between a TX mailbox of the Source Node and a RX Mailbox of the Destination Node. The same considerations applies to the " Data Sample Size (=N. of octets - 1) " in a Interrupt Message Packet.
h. In the Reply Packet should we reduce or not the copy of parameters from the corresponding Command Packet ? A complete copy is suitable to save a common format among reply packets and to provide additional security, but it is (slightly) paid in term of network bandwidth.
i. In the “Type/Command/Status/Sizes“ field of a Reply Packet, the Reply Status could be extended from 3 to 5 bits by replacing the 5 bit Command Code of the corresponding Command Packet (thus missing the copy of the Command Code in the Reply Packet); This makes the format of this field more similar in the 4 packet types; what about this possibility ?
j. Are there more suitable names to indicate each transaction ?
k. Are there more fitting names for the terminology used within this document ?