SpaceWire-D Baseline Considerations and Proposals

A.Tavoularis, N.Pogkas

TELETEL S.A., March 2011

1. SCOPE

This document was produced by TELETEL in the scope of the ESA/ESTEC contract "Evaluation Assessment and Prototyping of SpaceWire Protocols (SpW-D, RMAP)" CCN 03 to 22256/09/NL/CBI. Within this contract, TELETEL has performed analysis and simulation of the SpW-D draft protocol [2] on realistic network topologies. This document summarizes the main findings of these activities and provides baseline considerations and proposals for further standardisation of the SpW-D protocol.

2. INPUTS, REQUIREMENTS AND RESTRICTIONS

This section presents inputs, requirements and restrictions for SpW-D that have been proposed by SpW WG members and have been taken into consideration for the analysis and conclusions presented in this document.

2.1.1 Command and Control applications

Typical frequencies for synchronous Command and Control (monitoring and command) applications are the following:

- 1Hz (PPS driven services)
- 10Hz (AOCS, 2Hz, 8Hz, 16Hz, …, typical range for MIL-STD-1553B networks)
- 100Hz (high-end AOCS, pointing, maximum capability for MIL-STD-1553B networks),
- 1KHz (motor control, Robotics, high speed control loops, micro-vibration compensation, fine pointing)

Note: Control loops of 1KHz may be difficult to achieve in some network topologies because they need a large number of allocated time-slots per epoch interval for data transfers. (e.g. a 1KHz control cycle with 1ms period will require 1 every 8 time-slots for data transfer when time-code repetition is 122μs and 1 every 4 time-slots for data transfer when time-code repetition is 244μs).

2.1.2 High Throughput Data Transfer applications

- Network optimisation shall be driven by High Throughput Data Transfer (of typically payload data) in order to guarantee the throughput requirement of these traffic profiles (this approach has been presented in [1] section 3.4.2).

2.1.3 *Interoperability and Compatibility*

- In order to ensure interoperability, variability shall be minimized.
- Compatibility with existing devices (SpW routers and RMAP targets) must be considered, but not at the expense of bending the following fundamental requirements.

2.1.4 *SpW-D fundamental requirements*

- Determinism:
  - Transactions shall be scheduled, synchronised with Time Codes.
  - Transactions shall be time bounded, (executed within one time slot or a specific number of time slots in case multi-slot scheduling is used).
  - A transaction exceeding its schedule (one time-slot or the specific number of time-sots in multi-slot scheduling) shall be detected and aborted. Optionally Packet Routing timeouts and time slot usage monitoring would improve reliability.
- Strict Determinism:
  - A transaction failing to complete in the allocated time slot or in the allocated number of time slots in multi-slot scheduling shall not generate any perturbation in the next time slots.
- Integrity:
  - Data integrity shall be ensured from SpW and RMAP protocols (parity and CRC).
- Reliability:
  - Reliability shall be provided by the SpW protocol and the application, not by SpW-D.

2.1.5 *Time Codes propagation*

- Typical Time Codes' propagation delay shall be taken into account, which, in a complex network with cascaded routers, is less than 10 μs.
- Time Code interval must be large with respect to this delay/jitter, typically one order of magnitude larger (>100 μs). A safe margin or guard time at time-slot interval may be considered for this delay/jitter, as presented [1] section 4.3

3. PROPOSALS TOWARDS SpW-D STANDARDIZATION

3.1 SCHEDULING

3.1.1 Multi-Epoch Scheduling

Multi-epoch scheduling is a concept similar to the Major/Minor frames specified in MIL-STD-1553 and it is proposed as a promising scheduling method which satisfies the divergent requirements of low throughput C&C applications and high throughput DH applications. It can be used for cyclic monitoring/commanding above 1/(epoch interval) sec, as presented in [1], section 6.1.3. **A way to implement multi-epoch scheduling is through the use of the CCSDS time distribution proposed** during the 15th SpW WG. **The way time distribution and multi-epoch scheduling will be associated is still open**. An intuitive solution includes the Time Master transmitting Time Distribution frames periodically to each device (RMAP Target) in time-slot 0. From this epoch and until is it the turn of the same Target to receive the next Time Distribution frame, the Target keeps track of the epochs for its scheduling.

3.1.2 Multi-transaction Scheduling

The need for multi-transaction slots arises from the limited availability of time-slots in an epoch (due to the limited number of Time Codes) and their usage is proposed for C&C communication with small payloads, as presented in [1], section 6.1.1.

Two alternative solutions to be traded-off:

- Transmit a RMAP command immediately after the previous one. This results in more transactions per time-slot, but successive transactions to the same target are not supported. They will either be rejected (see 3.8) or will be executed after the Reply for the previous transaction has been transmitted nullifying the expected performance gain.
- Wait for reply and then issue the next transaction. Transactions to the same Target are supported but fewer transactions fit in one time-slot.

The analysis **proposes to use the first alternative** since:

- The complexity overhead for its implementation vs. the second solution is minimal and results in higher throughput.
- Successive transactions to the same Target in the same time-slot do not make sense (to be verified by the Steering Committee) since successive transactions are usually sent with Segmentation. Even if required, subsequent transactions may be mapped to different SpW-D channels and transmitted in other time-slots.

3.1.3 Multi-Slot Scheduling and maximum RMAP Payload length

Multi-slot scheduling is proposed in order support transmission of medium size data blocks (e.g. 1KB-4KB) which do not fit in a single time-slot without implementing segmentation. The maximum number of consecutive time-slots that shall be allocated for such transfers is **closely associated to the maximum allowed RMAP payload length**. The analysis performed proposes to **limit the maximum number of time-slots in multi-slot scheduling for the following reasons**:

- For **bounding the maximum latency**.
- For **implementation/cost issues**. Supporting Verified RMAP Writes implies that a verify buffer of the same capacity (**without taking EDAC into account)** is present at the target. Therefore, supporting large RMAP payload lengths may not be realistic. Implementing **a 16Kbytes verify buffer for example cannot be supported in the smallest FPGA of the ACTEL RTAX family** (54Kbits memory). It can be supported in the in the RTAX2000 device

(288 Kbits) but even in this case it would not be an optimum/cost effective use of the device (utilizes half the memory of a large/expensive device for the SpW-D protocol requirements).

- For verified writes the payload is first stored in the verify buffer, the CRC is validated and the payload is transferred to the Target's main memory before the transmission of the Reply. Consequently, for **large payloads the Reply transmission latency may increase prohibitively**.
- Large packets increase the probability of a multiple-bit error going undetected at the target. Although it is improbable to receive an erroneous RMAP packet with all the SpW NCHARs having correct parity and the RMAP CRC correct, **the RMAP 8bit CRC effectiveness has not yet been evaluated**.

The current analysis **initially proposes to limit the maximum RMAP payload for SpW-D to:**

- **4 Kbytes** for systems **implementing verified Write** transactions
- **16 Kbytes** for networks which **only implement non-verified Write** transactions

3.2 TIME CODE/SLOT PERIODICITY

The SpW steering committee has proposed to standardize the time-slot duration to either 122 µs or 244 µs. This is driven by the requirement to distribute CCSDS CUC frames and these values ensure that the Time frames are distributed periodically (**this infers a synchronization issue at system level presented in 3.9.2**) and their periodicity is adequate for S/C applications

Based on simulations for the proposed values (presented in [1], section 3.3.2.9) we have the following results:

Time code Interval 122 µs: $1/2^{13} = 1/8192 = 122.0703125$ µs ► 128 epochs per second

SpW Link Speed (Mbps)	Maximum Data Length (Bytes)	Notes
2	0	Cannot be used
10	56	Ok for C&C, too short for HTDT
50	432	
100	870	
200	1650	

Table 1: Maximum Data Length for 122µs Time-slot and various SpW link speeds

Time code Interval 244 µs: $1/2^{12} = 1/4096 = 244.140625$ µs ► 64 epochs per second

SpW Link Speed (Mbps)	Maximum Data Length (Bytes)	Notes
2	2	Too short
10	178	Ok for C&C, too short for HTDT
50	1018	
100	2000	
200	3760	

Table 2: Maximum Data Length for 244µs Time-slot and various SpW link speeds

Note: the maximum data length that can be achieved for a time slot interval is highly impacted by the delays imposed by the HW implementation. The simulation results presented above are based on the values for the HW delays specified in SpW-D Draft B [2].

3.3 Segmentation functionality

Implemented as a discrete layer above SpW-D/RMAP. The application should not have any visibility to the segmentation information. The proposed mechanism is to **implement Segmentation and Reassembly at the initiator side only** in order to keep **compatibility with existing RMAP Targets**. With the mechanism proposed in [1] section 6.3, the initiator segments the requests for the transfer of a SDU to multiple requests targeted to successive Target memory addresses. After the completion of a SDU Write transfer the segments are in sequence at the target memory. For read transfers **an open point is how the Initiator handles the Header and Payload fields of the Read reply**. If the Header and the Payload are stored in a contiguous Initiator Memory space then further actions are required for SDU reassembly. If they are stored in separate Initiator address spaces, then reassembly can be done by storing the payload of successive read replies in successive memory spaces. **An open point regarding SDU transfers towards a Target, is how the Target will be notified for the existence of a reassembled SDU in its memory**. Two alternatives have been identified:

- Implement logic through the Target Authorization Logic Interface. This solution is **not proposed** as it requires:
    - Using different Protocol IDs for normal RMAP transactions and Segmented Transfers.
    - Complex Authorization Logic at the Target in order to decode the Segmentation/Reassembly information encoded in the TID field of the RMAP transactions.
- Address this issue at system level. This is the **proposed solution** as it does not require any changes at the target side at the cost of lower performance.

This solution is **compatible with all scheduling alternatives and with the proposed Retry functionality** (see 3.4 since there is no possibility for out-of-order segments delivery at the Target).

3.4 Retry functionality

Retry functionality is not proposed at SpW-D level. It is proposed to implement retry at a higher level. The implementation of Retry without segmentation is straightforward. However for Retry of segmented SDUs the current analysis proposes:

- At the initiator side if the RMAP reply associated with a segment of a SDU indicates error or times out the rest segments are aborted
- At the initiator side the upper layer shall be immediately informed about the failure in the transfer of the SDU in order to take the appropriate actions.
- At the target side no actions shall be taken since subsequent proposal in this document prohibits the usage of FIFO interface at the target.
- It is recommended to use a structured TID (including Sequence Number and Target channel) in order to provide for identification of the Target denying the transfer and the failure cause (e.g. if the transfer fails at the same SQ it may indicate that the attempted transfer exceeds the specific target's memory space) and to provide for partial SDU Retry in the case of a failed SDU Read Transaction.

3.5 Priority and Channels

Priority can be added by introducing the concept of channels in SpW-D. With this approach one or more channels are associated with a certain destination. The following advantages are offered:

- By using multiple channels to a single destination, flows of different priority can be set-up.
- Issuing non-successive RMAP commands to the same Target in a multi-transaction slot can be supported.
- In addition, schedule table implementation is simplified since:

- A single bit is used per Target for each time-slot instead of 8-bits (Target Address).
- The address space formed by the Targets to which an Initiator issues commands is not contiguous.

3.6 Timing Faults Containment

Functionality similar to the KILL functionality is proposed for implementation at the Initiator side. Furthermore, Timing faults containment shall be performed within the timeslot(s), as presented in [1], section 4.4.

Note: Even with this method, **determinism is not ensured in case of babbling idiots**. This issue can be addressed by the SpW routers.

3.7 Determinism vs. Strict Determinism

Since it is proposed not to implement Retry at SpW-D level (see 3.4), a failed transaction cannot have any impact on the scheduling of the subsequent transactions and therefore **strict determinism is implied**.

3.8 FIFO Interface – Non Incremental RMAP addressing

The non-incremental memory addressing of RMAP implies a receive FIFO at the target. In case the target host application (e.g. a processor) does not read the RMAP command on time this may cause blockage of the SpW path from the initiator to the target in case another command is received due to lack of FCTs at SpW level. The latter can be overcome by rejecting RMAP commands if the Target is not idle, thus not retaining compatibility with existing devices.

Incremental memory addressing RMAP commands are issued towards Targets with a linearly addressed target memory and this problem is mitigated. In a similar scenario, the data at the target memory is overwritten before being read by the target application, but the SpW path links are not blocked.

To this respect and in order to simplify the Retry mechanisms for the upper layers (as shown in the analysis above) **the current analysis proposes not to use the non-incremental memory addressing**.

3.9 System/Network Level Issues

3.9.1 SpW Link Speed

It is proposed to use a **uniform link speed for all links in the network**, since Time-Code propagation and consequently Time Slot boundaries may be perceived differently by Initiators receiving them through links of different speeds and this will make network-wide traffic analysis more difficult. It is also recommended to set the minimum link speed over 10Mbps which is the default speed for link initialization. Usage of lower speed is not recommended since:

- Adopting the 122 μs and 244 μs time-slots means that practically no payload can be transmitted in 2 Mbps links.
- The gains in power consumption by reducing the link speed to 2Mbps are not significant since the SpW uses LVDS signalling.

Note: This proposal has not been examined from other points of view (e.g. EMC/EMI)

3.9.2 Time Distribution

If the intent of proposing time-slot periods which is to result in S/C-wide synchronous network operation then the way **clock timing is distributed within the S/C shall be taken into account**:

- If the Time Master (the one distributing the Time Distribution CCSDS frames) is coherently clocked (e.g. with the use of a PLL) with the Time-Code master then there is no jitter in the transmission of Time-Code in the network.
- In case the two devices are not coherently clocked then synchronization of Epochs with the CUC time is not achieved, since it is the Time Codes master who determines when the CUC frames are transmitted. If, **for example,** the two devices use N ppm crystals and the CUC frame is transmitted to the Time Codes master every M seconds, then the maximum skew in the time kept in the two devices is 2NxM seconds. In order to synchronize the Epochs with the CUC time the Time Codes master shall:
  - Change the duration of a certain Time Slot in order to compensate the 2NxM time skew possibly resulting in a Time Slot in which **RMAP command(s) may not fit**.
  - Compensate the time-skew in the next M seconds in order to distribute the difference in all Time Slots of the next epochs, which is an **extra processing overhead for the Time-Codes master**.

3.9.3 Initiators Scheduling and Traffic Profiles

Traffic sources must be synchronised with time-slot/epoch interval. If the inter-arrival time of a traffic source is not an integer multiple of time-slot interval there will be waste of bandwidth due to allocated time-slots for data transfer with a period less than the required inter-arrival time (this was required to not cause transmit buffer overflow at traffic sources).

This was encountered in the simulation scenarios of [1] section 3.4, where the instruments generated science packets with different inter-arrival times that could not “match” time-slot duration or epoch interval. As a result, more time-slots were allocated for these transfers than actually needed, resulting in a maximum bandwidth for the traffic flows larger than the required by the actual traffic profile. In this case there will be allocated time-slots in some epochs where the instruments will not have data to transmit, wasting network resources.

4. OPEN POINTS

4.1 OPEN POINTS

4.1.1 CCSDS Time Distribution and Multi-epoch Scheduling

Regarding the CCSDS Time Distribution the following points shall be addressed:

- Define if only the CUC format will be supported for Time Distribution.
- Define if a standard Target memory address will be specified for writing the Time Distribution frames.
- Define if standard time-slot(s) for each epoch will be used for Time Distribution.
- Define the required Time Distribution refresh rate for each SpW-D Initiator.
- Define how does Time Distribution and Multi-epoch scheduling are associated.
- Define how the skew between the Time Distribution master and the Time Codes master shall be compensated.

4.1.2 Layers representation

The top to bottom order of the SpW-D layers according to SOIS should be Segmentation, Retry, RMAP, Resource Reservation (Scheduling). **In a HW implementation however the boundaries may not be so discrete** since this layer stack has the following problems/drawbacks:

- According to this proposal, Segmentation will be a part of SpW-D, but **Retry will be implemented at higher layers** which means that Retry will be implemented above segmentation.
- According to this proposal **Retry at segment level is not supported** so there is no reason for Segmentation to be above Retry.
- In a HW implementation, implementing Scheduling below the RMAP implies that one queue will be instantiated for each SpW-D Channel/Target Address which **results in a non-optimum HW implementation**.

4.1.3 Segmentation

Regarding the Segmentation the following points shall be addressed:

- Define how a Target will be notified for the existence of a SDU in its memory.

4.1.4 Support for PTP

Regarding the support for Packet Transfer Protocol the following points shall be addressed:

- Define how the PTP packets are encapsulated in RMAP commands.
- Define how a Target will know that a PTP is contained in a RMAP Write command. By using different PID? By encoding it in the TID field? By performing writes to a specific Target memory address?
- Define how a Target will be notified for the existence of a PTP packet in its memory (a receive indication is required by the PTP protocol and SOIS packet transfer service)

5. SUMMARY

The following Table summarizes the proposals and constraints for SpW-D protocol.

Category	Proposal
Supported Scheduling alternatives	Simple
	Concurrent
	Multi-slot scheduling May be used to avoid segmentation for medium size data blocks
	Multi-transaction May be used to maximise slot usage for C&C with small payloads
	Multi-epoch (associated with CCSDS Time Distribution) May be used for cyclic monitoring/commanding above 1/(epoch interval) sec
Time Code interval	122μs or 244μs
Maximum allowed RMAP payload	4Kbytes if verified Writes are implemented 16Kbytes if only non-verified Writes are implemented
Supported RMAP Command options	Incremental addressing Supported Non-incremental addressing **NOT** supported
Segmentation and Reassembly	Optional Implemented at the Initiator side only
Retry	**NOT** implemented at SpW-D level
Priority	At the Initiator side Supported through the concept of channels
Determinism Type	Strict determinism
SpW Link Speeds	Uniform in the whole network Minimum SpW link speed 10Mbps

REFERENCES

[1] "TN-101: SPW-D Implementation Specification rev.0.1", Evaluation Assessment and Prototyping of SpaceWire Protocols (SPW-D, RMAP) (CCN3 to ESTEC contract 22256/09/NL/CBI).

[2] "SpaceWire-D – Deterministic Control and Data Delivery over SpaceWire Networks", Draft B, April 2010 (Ref: SpW-D Draft B).

[3] CCSDS Unsegmented Code Transfer Protocol (CUCTP), Marko Isomäki, Sandi Habinc, Aeroflex Gaisler AB, SpW conference, 10 October 2010.