PROTOTYPE IMPLEMENTATION OF A ROUTING POLICY USING FLEXRAY FRAMES CONCEPT OVER A SPACEWIRE NETWORK

Sev Gunes-Lasnet(1), Olivier Notebaert(1)

(1) *Astrium, 31 avenue des Cosmonautes, 31402 Toulouse Cedex - France*
Phone: +33 (0)5 6219 7663 - Fax: +33 (0)5 6219 7158
Email: sev.gunes-lasnet@astrium.eads.net

ABSTRACT

The benefit of SpaceWire as an efficient data-link to transmit science data on board spacecraft is now demonstrated through the growing development of SpaceWire in major space science projects such as Bepi-Colombo, Gaia and the James Webb Space Telescope. The efficiency of SpaceWire becomes even more obvious when used in a network configuration. Although not yet really popular onboard spacecraft, a network configuration not only reduces the overall data links mass, it also enables flexible implementation of scalable distributed systems which can be of great interest for future applications. However, state of the art SpaceWire networks cannot offer sufficient levels of communication services quality with the existing protocols which limits their field of application to non critical applications from the dependability or real-time standpoint.

The FlexRay communication concept developed in the automotive industry takes into account dependability and real-time properties to provide guaranteed access to the network for critical data transfers through parallel channels and time slotting. The remaining bandwidth is shared on a best effort basis for non-time critical data transfers. A study performed at Astrium makes an adaptation of FlexRay protocol elements such as communication cycle and routing strategy over a SpaceWire network.

1. CONTEXT

Several drivers for onboard data links and I/Os upgrade can be identified, such as new missions requiring autonomous manoeuvres for space exploration or the development of robotics in space. The emergence of new spacecraft control mode, e.g. using vision and star trackers instead of accelerometers and gyroscopes, induced new needs in terms of data processing and in particular for the on-board data links capabilities [1] [2].

As a consequence, enhanced capabilities are required for on-board data links in particular due to the increase of data exchange volume (towards the Gigabits/s range), the real-time command and control requirements and the need for in-orbit software maintenance, together with a need for lower power consumption to perform solar system exploration and to allow to reach further spacecraft autonomy.
Moreover, in the context of optimisation of on-board data links to satisfy to new needs in terms of data processing, there is a growing interest for decentralised/distributed architectures implying peer-to-peer communications rather than master/slave ones.

In order to face the identified challenges for on-board data processing and in particular for on-board data links in an effort of cost reduction, the re-use rate, maturity and interoperability of the technology developments should be optimised; this is why standardisation is pushed forward by the different space agencies and industries [3].
Other constraints such as harness optimisation are pushing for network or buses configurations to simplify the integration and operations of spacecrafts.

2. ON-BOARD DATA LINKS OPTIMISATION

2.1. Current use of high speed data links

1355 links, which are the precursors of SpaceWire, are used in several spacecrafts such as Science data to Mass Memory (Cryosat, Rosetta, Mex/Vex) and for telecom signal dynamic switching (Inmarsat4).
Other high data rate links are used for Gbit performance requirements in missions such as Pleiades, TerraSAR-X. High speed data links are also used as on-board computer ground test interface for software instrumentation.
SpaceWire has been considered in most studies on future data processing architecture and Leon System-on-Chip prototypes:

- In ESA studies such as SCoC and the A3M demonstrator, A3SysDef, Gamma, Disco…[4] [5][6][10]
- And also in national agencies and EADS-Astrium internal projects such as ALF3, Unionics, MAEVA, PADAPAR… [7] [8]

SpaceWire/ECSS-E50-12 supports the need for high performance data processing on future spacecrafts and is recommended for any point-to-point high speed link with a data rate fewer than 200 Mbps; it is viewed as a future solution towards more generic payload data processing systems through on-board data networks [9]. SpaceWire networks could also be extended to the whole data processing system (e.g. for small vehicles, robotics…).

2.2. On-Board data links optimisation strategy

The current typical on-board data links architecture is challenged by new needs in particular in terms of Payload data processing, as described in the first paragraph of this paper.

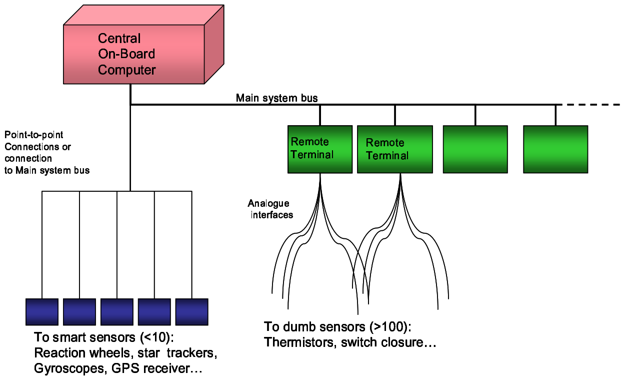


Figure 1: Typical on-board data links architecture

The optimization of on-board data links targets the use of on-board networks, in particular for payload data processing. On-board networks could answer the anticipated needs in terms of data throughput, determinism and reduced harness mass. SpaceWire is in this context considered to be of high interest for the development of payload data handling building blocks, in particular because it could act as a multi-node backbone, see Figure 2.

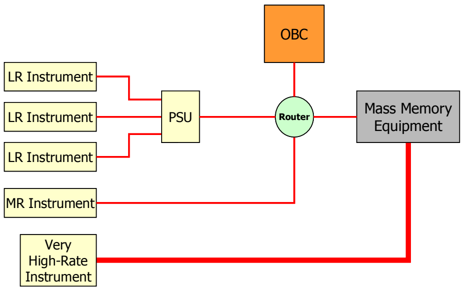


Figure 2: SpaceWire as a multi-node backbone

Such SpaceWire networks are considered to be able to optimise on-board resources and performances; to insure their development, the consolidation of identified points presented in the following paragraph is necessary.

2.3. Critical elements for SpaceWire networks development

As identified by the SpaceWire working group and discussed during the first SpaceWire International conference in 2007, critical aspects for the development of SpaceWire networks on-board spacecraft are in particular:

- The availability of SpaceWire Network building blocks elements for embedded systems such as space qualified components and IP's for integration in System on Chips;
- Ground support software/hardware, e.g: adapters to ground networks (usb, ethernet…), Traffic monitor/simulator, architecture model, network administration SW, simulation…
- Standardised protocols ensuring time deterministic packet delivery, (e.g. derived from existing aeronautics standards).

The activity described in this paper focuses on the deterministic protocols issue.

3. PROTOTYPING AND TESTING OF A ROUTING POLICY USING FLEXRAY FRAMES CONCEPT OVER A SPACEWIRE NETWORK

High level protocols implemented over SpaceWire will allow ensuring the respect of timing constraints, while respecting the SpaceWire standard.

To perform a first iteration analysis of how a high level protocol could be implemented over SpaceWire, the prototyping of concepts from the automotive protocol FlexRay has been performed under the form of a pilot implementation over a SpaceWire network. This implementation will allow further analysis of the concept and in particular performance assessment.

3.1. FlexRay consortium overview

The Flexray consortium originally chartered in end of December 2006, with a new charter established in January 2007. It targets the automotive market, with automotive control applications.

The core members are BMW, Bosch, DaimlerChrysler, FreeScale, GM, Philips and Volkswagen. The consortium is composed of 7 Core, 13 Premium, 47 Associate, and 27 Development members so far.

Initially, the consortium was only open to automotive industry players (semiconductor manufacturers, equipment and system suppliers, tools and services suppliers and manufacturers). However in late 2006, the consortium agreed the membership to EADS Innovation Works Germany as a move to non-automotive companies. Technology use and licensing is still not cleared for non-automotive applications.

3.2. FlexRay technology overview

FlexRay's main characteristics are directly resulting from the automotive X-by-Wire requirements: fault-tolerant clock synchronization via a global time base, collision-free bus access, guaranteed message latency, message oriented addressing via identifiers, scalable system fault-tolerance, and speed of one order of magnitude higher than CAN [11].

There are two interesting properties of FlexRay for supporting the deterministic communication and reliability required by space systems: the first one is the division of the bandwidth between a deterministic time triggered communication part and a user defined run-time scheduled communication part; the second property is the possibility of using redundant channels if this is desirable.

FlexRay uses the OSI-model as reference model. The layers that FlexRay uses are the Application, Data Link and Physical layers (Figure 3). The Network and Transport layer are typically placed in the second layer, and the Session and Presentation layers in the seventh layer.

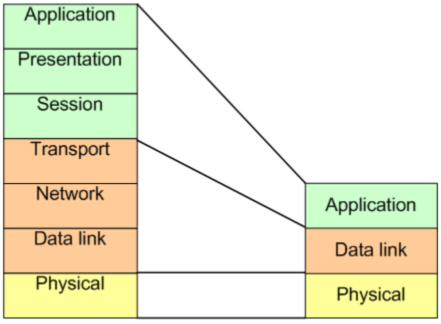


Figure 3: Layer mapping between OSI and FlexRay

3.3. Reuse of automotive protocols concepts

Given the position of the FlexRay consortium towards non-automotive applications, the prototyping implementation of a routing policy using its concepts has been performed, rather than the direct evaluation of fully supported FlexRay features products.

3.4. Prototyping platform

The prototyping platform chosen is depicted in Figure 4: It is composed of a SpaceWire router from 4Links, a DSI (Diagnostic SpaceWire Interface) also from 4Links which has been programmed for analyzing the SpaceWire traffic, and 3 boards with SpaceWire ports and LEON2 processors in FPGAs. Two boards are AVNET boards, while the last one, acting as time master, is a Pender board. Two SpaceWire ports have been used on the AVNET2 board (see Figure 4).

Moreover, a SpaceWire IP tunnel [10] from Star Dundee has been used between the router and the different boards to monitor the traffic and check for eventual communication problems. The DSI has been used to analyze the traffic which arrives at its ports. On the contrary of the router, it is possible to control the DSI by programming the API provided by the constructor.

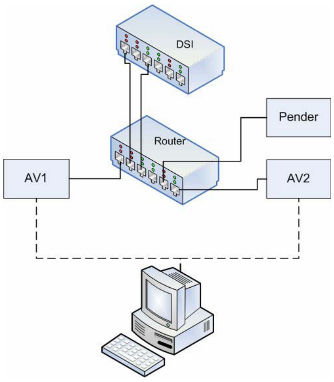


Figure 4: prototyping platform

3.5. Communication model prototyping

The Rosetta spacecraft on-board communication model has been chosen in order to allow the simulation of real payload data to be transmitted over the SpaceWire network. Within Rosetta data handling architecture, 8 modules use IEEE-1355 serial links. The traffic chosen for this pilot implementation has been derived from the traffic forecasted on the 1355 links between the Solid State Mass memory and the avionics processor, between one instrument and the processor, and between one instrument and the memory.

During the prototyping, a time master on the network distributes a global time reference; this is the Pender board task. The different modules use this time reference for sending packets over the network.

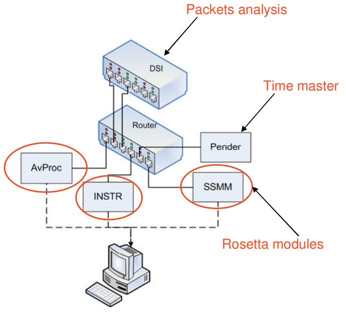


Figure 5: Prototyping configuration

The Pender board has one six-bit time counter (0 to 63), which is incremented each second. The generated time code signal is broadcasted to all the output ports of the router. All nodes connected to the router (all AVNET boards) have one six-bit counter. When the interface receives the broadcasted time-code, it updates its associated time-counter with the new time. This common time reference is used for the transmission of the packets. Each board has been programmed to transmit packets which are based on the Rosetta payload packets: One board acts as Avionics processor board, while the other corresponds to the Instrument. Their flow diagram is depicted in the following figures.

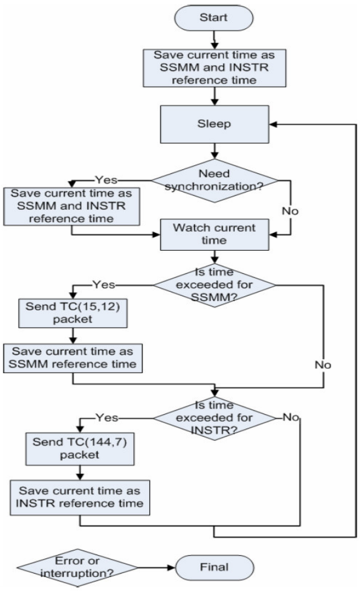


Figure 6: Flow diagram of the avionics processor board send packet task

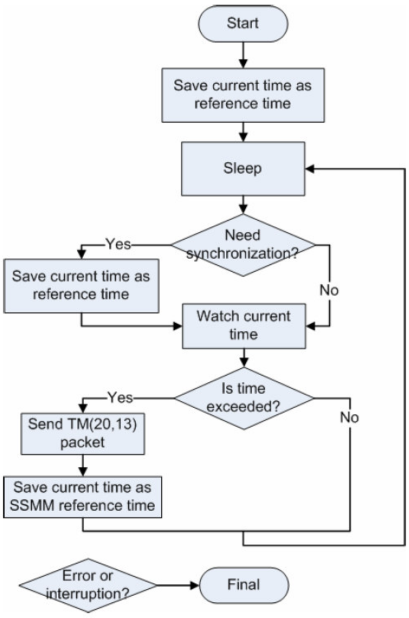


Figure 7: Flow diagram of the instrument board send packet task

The configuration implements a priority based bandwidth reservation: The packets with high priority are guaranteed to be transmitted first during the static part of the cycle. In order to re-use concepts from the automotive, two time zones have been defined for each communication cycle, as depicted in Figure 8;

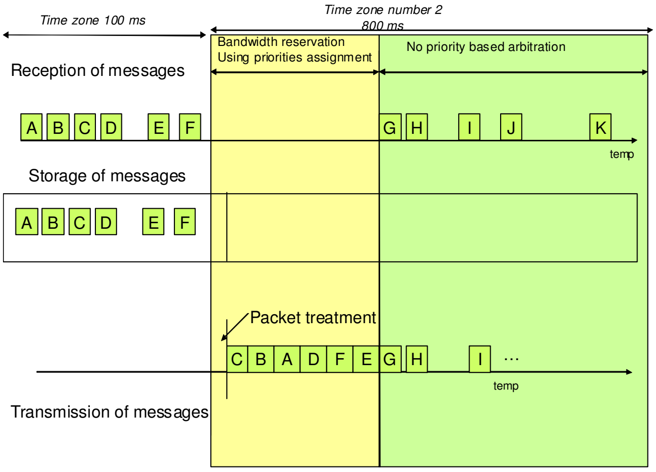


Figure 8: Communication cycle

A communication cycle of 900ms has been defined by dividing it into a priority time frame of 100ms and a secondary time frame of 800ms:

- The first time zone takes as reference the first packet received, and lasts until 100ms after the reception of the first packet. The packets which arrive in this time zone will be considered as simultaneously arrived, and they will be managed in base of their priorities. This is considered to be the priority zone, and is designed to implement similar capabilities as the time-triggered part of the communication cycle. Due to the lack of synchronization between the different modules in the pilot implementation, this time zone can however not be considered as fully time-triggered.
- The second time zone takes as reference the first packet received after the end of the priority zone and lasts 800ms. In this time zone, there is no priority based packet processing; the packets are managed with the original SpaceWire driver. This time zone has been implemented with the aim of allowing event driven communications during each communication cycle.

Such a communication cycle model has been chosen in order to correspond to the identified needs of bandwidth reservation and event triggered packets transmission over a SpaceWire network for payload data processing.

The aim of the pilot implementation was to implement a first iteration of a communication cycle design over a SpaceWire network rather than to implement a full time-triggered approach; the routing is done through an analyse-and-transmit routing approach. The prototyping platform will allow further analysis and performance assessment of high level protocols over SpaceWire networks.

3.6. Packet analysis

The DSI acts as packets analyser unit, as depicted in the flow diagram in Figure 11, and has two of its ports connected to the router. The priority packets are stored during the 100ms time slot and once the allocated time is over, a packet treatment thread sorts the messages by descending priority (inferior number, higher priority). The packets are then sent one after the other, after which the secondary time frame of 800ms is started. Finally, the message list is deleted. The ports and priorities used for each link are specified in Figure 9.

	Port	Port DSI	Port exit DSI	Priority
AV1	7	3	2	7
AV2.0	5	2	2	5
AV2.1	8	2	2	8

Figure 9: ports and priorities used

For this prototyping, the priority has been introduced in the second octet of the SpaceWire packets as depicted in Figure 10.

<table>
<tr><td colspan="2">Destination id</td><td colspan="2">Cargo</td><td colspan="2">End of packet</td></tr>
<tr><td colspan="6">SpaceWire packet format defined in ECSS SpaceWire standard</td></tr>
<tr><td>Destination id</td><td colspan="2">Priority</td><td colspan="2">Cargo</td><td>End of packet</td></tr>
<tr><td colspan="6">SpaceWire packet format proposed by this communication model</td></tr>
</table>

Figure 10: SpaceWire packet format used in the pilot implementation

The flow diagram in Figure 11 shows how the analysing task manages the packets.

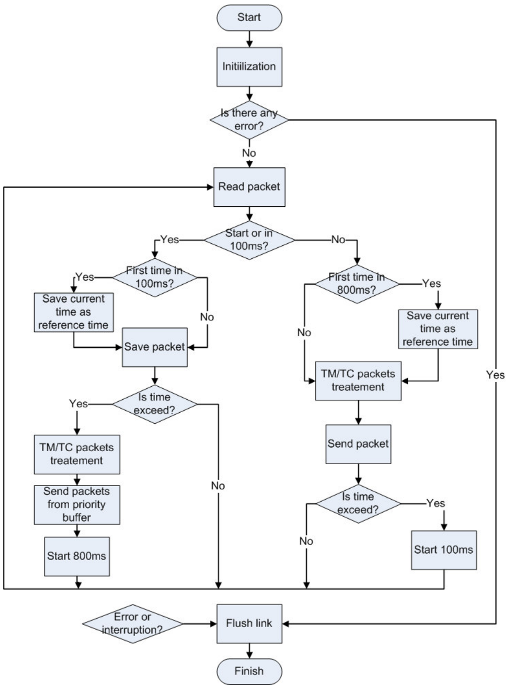


Figure 11: Flow Diagram of the DSI function as packet analyser, using Rosetta packets

4. PROTOTYPE TESTING RESULTS ANALYSIS

The reorganisation of the packets during the first part of the communication cycle has been monitored, as well as the routing of packets without priority based bandwidth attribution during the second part of the cycle. Figure 12 presents an overview of the results obtained for the priority based treatment of the Rosetta-like packets:

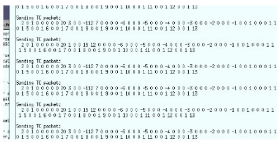
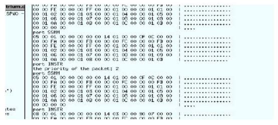
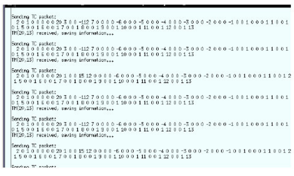
 Figure 43: AvProc sending TC messages	The AvProc module sends TC packets.
 Figure 44: The messages sent by the DSI after the treatment	The DSI receives the packets, it treats them, and then the packets are sent to the logical port.
 Figure 45: the AvProc receiving TM messages	The AvProc receives a TM packet. That means that it has received an answer to a request from another module.

Figure 12: Rosetta payload packets simulation overview

The transmission of packets and its correct processing by the programmed DSI has been observed during a test procedure which parameters are presented in Figure 13.

Parameters of the test	*values*
Number of messages transmitted	1 message each 100 ticks
Duration of the test	More than 100 messages sent
Number of messages received in time frame 1	Average of 5 messages
Number of messages received in time frame 2	Average of 40 messages

Figure 13: Test procedure parameters

The communication cycle concept presented in paragraph 3.5 has been successfully implemented; during the first part of the communication cycle, each board transmits different packets which are re-organised by the DSI and routed by the SpaceWire router. During the second part of the cycle, no priority based routing is used, and the DSI does not re-organise any packets arriving at its input ports.

Several improvements can be made to the prototyping developed for this pilot implementation. It will be needed in another iteration of the prototyping to increase the traffic between the modules. A higher number of modules can be used to match the Rosetta configuration. For this implementation, default speeds of 100 mbps have been used for the SpaceWire links, this can be adapted in further iteration.

In order to perform a further test of a bandwidth reservation policy based on priorities, a synchronisation method between the different modules clocks is needed. This will insure precise attribution of communication slots to each node connected to the SpaceWire network.

The optimisation of the communication cycle definition needs to be performed; the length of the packets and of the communication cycle can be adapted.

In addition, for an implementation based on the FlexRay concept, bandwidth reservation relies on the intelligent routing strategy, which is located in the router. The DSI has been used as an intelligent router in order to ease the implementation given that the DSI could easily be programmed. Such intelligent routing strategy requires the specification of dedicated router capabilities such as slots reservation for each user on the network, control of the packets transmission to prevent babbling idiot errors, and possible further capabilities.

5. CONCLUSIONS AND PERSPECTIVES

The prototyping presented is a first implementation of concepts developed in the automotive industry, which are of interest for the development of SpaceWire networks building blocks for future on-board payload data processing. The implemented model is to be further iterated in particular in terms of synchronization, communication model optimization and representative traffic.

Further developments are on-going; the prototype will be used for the measurement of the sensitivity of the latency with respect to the length of packets transmitted over the network, and for the optimal dimensioning of the network elements: Size of an internal router buffer, optimal length of the communication cycle…

Moreover, the platform used for this pilot implementation can be used for the evaluation of further concepts of real time protocols over SpaceWire networks, thus contributing in the development of future on-board applications using SpaceWire networks.

6. REFERENCES

[1] B. Frapard, S. Mancuso, Astrium Satellites “Vision Navigation for European Landers and the NPAL Project”, *Proceedings of the 6th ESA International Conference on Spacecraft, Guidance, Navigation and Control Systems, 2005*

[2] “Autonomous Vision-based navigation demonstration », Bernard Polle, Anthony Villien, Jacques Lheritier, Benoit Frapard, Astrium Satellites, *Proceedings of GNC 20087th International ESA Conference on Guidance, Navigation & Control Systems*

[3] “Benefits of the standardisation efforts for on-board data interfaces and services », Olivier Notebaert, Astrium Satellites, *Proceedings of SpaceOps 2006 conference*

[4] Marc Le Roy - Astrium Satellite, “Advances Avionics Architecture and Modules (A3M) Final Report”, ESA Contract 13024/98/NL/FM (SC), September 2003.

[5] Philippe Chevalley-ESA/ESTEC, Luc Planche, Astrium Satellites, Stuart Fowel-Scisys, DisCo – A Space-Oriented Middleware for future Payload Data Handling Equipments Proc. ‘DASIA 2006 – DAta Systems In Aerospace Conference’, Berlin, Germany, 22-25 May 2006 (ESA SP-630, July 2006).

[6] Christophe Honvault - Astrium Satellite, “Generic Architecture for Mass Memory Access (GAMMA) executive summary” ESA Contract 17437/03/NL/AG.

[7] « MAEVA for very integrated, low power plat-form systems », Marc Lefebvre, Jean-François Coldefy , Astrium Satellites, *Proceedings of DASIA 2005*

[8] Christophe Honvault, Olivier Notebaert – Astrium Satellites, “Prototyping a Modular Architecture for On-Board Payload Data Processing - PADAPAR”, *Proc. ‘DASIA 2007 – Data Systems In Aerospace Conference’, Naples, Italy, 29 May - 1 June 2007 (ESA SP-638, August 2007).*

[9] Christophe Honvault, Olivier Notebaert - – Astrium Satellites, SPACEWIRE NETWORKS FOR PAYLOAD APPLICATION*, International SpaceWire Conference 2007, Dundee, United Kingdom, 17-19 September 2007*

[10] Christophe Honvault (Astrium Satellites), Raffaele Vitulli (ESA/ESTEC) Topnet Pilot Operation Implementation: A First Step To Virtual Satellite Integration, *Proc. ‘DASIA 2008 – data Systems In Aerospace Conference’, Palma de Mallorca, Spain, 27-30 May 2008 (ESA SP-665, August 2008)*

[11] “Flexray - An ANSWER TO THE CHALLENGES FACED BY SPACECRAFT ON-BOARD COMMUNICATION PROTOCOLS” Sev Gunes-Lasnet, Gianluca Furano, *Proceedings of DASIA 2007 conference*