# SPACEWIRE-RT

**Session: SpaceWire Standardisation**

**Long Paper**

Steve Parkes and Albert Ferrer

*School of Computing, University of Dundee, Dundee, DD1 4HN, Scotland, UK.*

*Email: sparkes@computing.dundee.ac.uk.*

## ABSTRACT

SpaceWire-RT (SpaceWire Reliable Timely) aims to provide a consistent quality of service mechanism for SpaceWire and to support proposed CCSDS Spacecraft Onboard Interface Services (SOIS) sub-network services. Furthermore it is intended to support control applications running over SpaceWire where timely delivery of information is essential. To achieve this SpaceWire-RT implements quality of service mechanisms over SpaceWire. SpaceWire-RT forms the quality of service layer of a complete SpaceWire protocol stack which incorporates the Remote Memory Access Protocol (RMAP) and other related protocols currently under development.

This paper introduces SpaceWire-RT, describes how SpaceWire-RT fits into the envisaged SpaceWire protocol stack, outlines the proposed quality of service mechanisms, provides an initial service interface definition for SpaceWire-RT and gives a summary of its architecture.

## 1 INTRODUCTION

SpaceWire [1] is designed to connect together high data-rate sensors, processing units, memory sub-systems and the down link telemetry sub-system. It provides high-speed (2 Mbits/s to 200 Mbits/s), bi-directional, full-duplex, data links which connect together SpaceWire enabled equipment. Networks can be built to suit particular applications using point-to-point data links and routing switches. The remote memory access protocol (RMAP) [2] was subsequently designed to provide a simple, standard means for one SpaceWire node to write to and read from memory inside another SpaceWire node.

SpaceWire and RMAP operate with a best effort quality of service. While error detection, reporting and recovery techniques are defined in both standards, there is no defined means of recovering any data that was lost or that arrived at its destination in error. Also there is no concept of timeliness in either of these standards. For many SpaceWire applications this is not a problem, but for other applications quality of service is a key issue.

SpaceWire RT [3,4] provides a consistent quality of service framework for SpaceWire and supports both the proposed CCSDS SOIS sub-network services [5, 6, 7, 8] and control applications running over SpaceWire.. To achieve this SpaceWire-RT implements quality of service mechanisms over SpaceWire.

## 2 COMMUNICATIONS MODEL

It is important to understand the communication model used by SpaceWire and SpaceWire-RT.

### 2.1 SPACEWIRE

SpaceWire uses point to point links to directly connect one node to another node. SpaceWire networks can be constructed using wormhole routing switches. Nodes can then be indirectly connected via one or more routing switches.

SpaceWire provides a stream service. The inputs and outputs to a SpaceWire point to point link are FIFOs. Data presented to an input FIFO is transferred across the SpaceWire link to the other end where it appears in and can be read from an output FIFO. A link level flow control mechanism is defined in SpaceWire which prevents overflow of the output FIFO. When the output FIFO becomes full no more data can be transferred across the SpaceWire link until some data is read out of the output FIFO making space available.

There is no management of memory at the destination i.e. there is no mechanism in SpaceWire to reserve memory or other resources in the destination node. If the destination user decides to stop reading data from an output FIFO the corresponding SpaceWire link will block.

### 2.2 SPACEWIRE-RT

SpaceWire-RT aims to provide a quality of service layer for SpaceWire that gives SpaceWire reliability and timeliness properties. The intention is for this to be done in such a way that any application that used SpaceWire would be able to run over SpaceWire-RT and gain benefit from the QoS provided. This application would be able to run with other applications which required improved QoS (either reliability or timeliness or both) without impairing the QoS of those other applications. To be able to run existing SpaceWire applications over SpaceWire-RT the interface to SpaceWire-RT has to be conceptually the same as that of SpaceWire. The communications model for SpaceWire-RT is thus one of virtual point-to-point connections across a SpaceWire network (referred to as channels) each of which connects a source channel buffer (input FIFO) in one node to a destination channel buffer (output FIFO) in another node.

SpaceWire-RT adds QoS to the SpaceWire paradigm, by providing a stream service over virtual point to point links. Any SpaceWire packet can be sent over a SpaceWire-RT virtual point to point link (channel) receiving the QoS of that channel.

SpaceWire-RT does not provide management of user memory or other resources. It only manages the source and destination buffers (FIFOs) within SpaceWire-RT. Management of these buffers by SpaceWire-RT ensures that no SpaceWire packets are left strung out across the SpaceWire network blocking other traffic while waiting for space in a destination buffer.

## 3 SPACEWIRE PROTOCOL STACK

SpaceWire-RT is part of the layered protocol stack for SpaceWire which is illustrated in Figure 1.

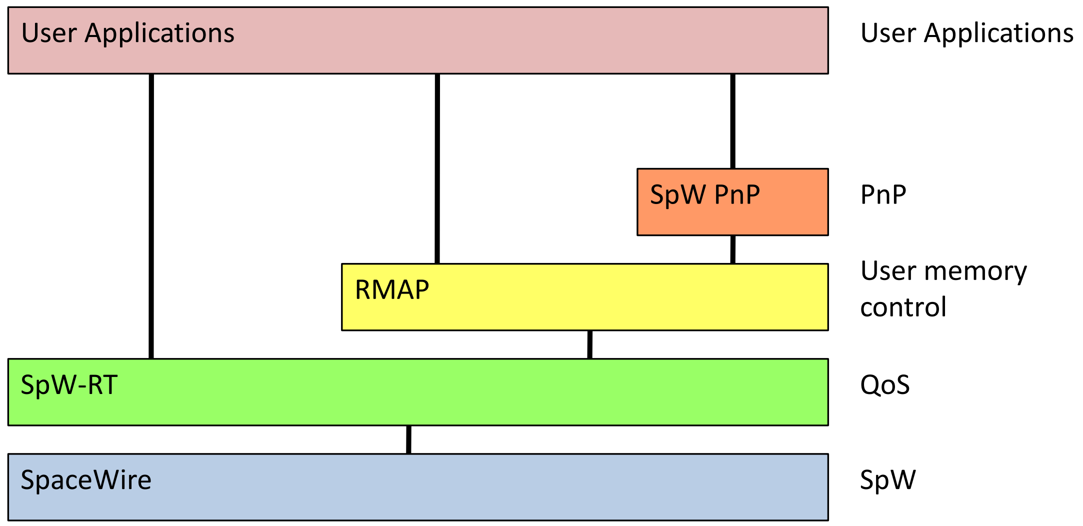


**Figure 1 SpaceWire Layered Protocol Stack**

SpaceWire is at the bottom of the protocol stack sending SpaceWire packets across the SpaceWire network from source to destination. Immediately on top of SpaceWire is SpaceWire-RT providing quality of service. All traffic has to pass through SpaceWire-RT otherwise the reliability and timeliness QoS cannot be ensured. User applications can talk directly to SpaceWire-RT. RMAP provides a mechanism for reading from and writing to memory in a remote node. SpaceWire-PnP (plug and play) uses RMAP for configuration and management of nodes on the SpaceWire network. User applications can use the services provided by RMAP or SpaceWire-PnP as well as talking directly to SpaceWire-RT. All use the same, consistent quality of service framework.

The CCSDS Spacecraft Onboard Interface Services (SOIS) working group has defined a set of common communication services for use onboard a spacecraft. The SOIS subnetwork layer and three of the services provided are illustrated in Figure 2. The SOIS Packet Service provides for delivery of packets across a subnetwork, the Memory Access Service for the access of memory devices on the subnetwork, and the Device Discovery Service supports plug-and-play capability with notification services.

RMAP and SpaceWire-PnP provide the Memory Access Service and Device Discovery Services. The SOIS Packet Service is provided by a SpaceWire Packet Transfer Protocol (SpaceWire-PTP) that sends packets across the SpaceWire network, providing buffer management and flow control.

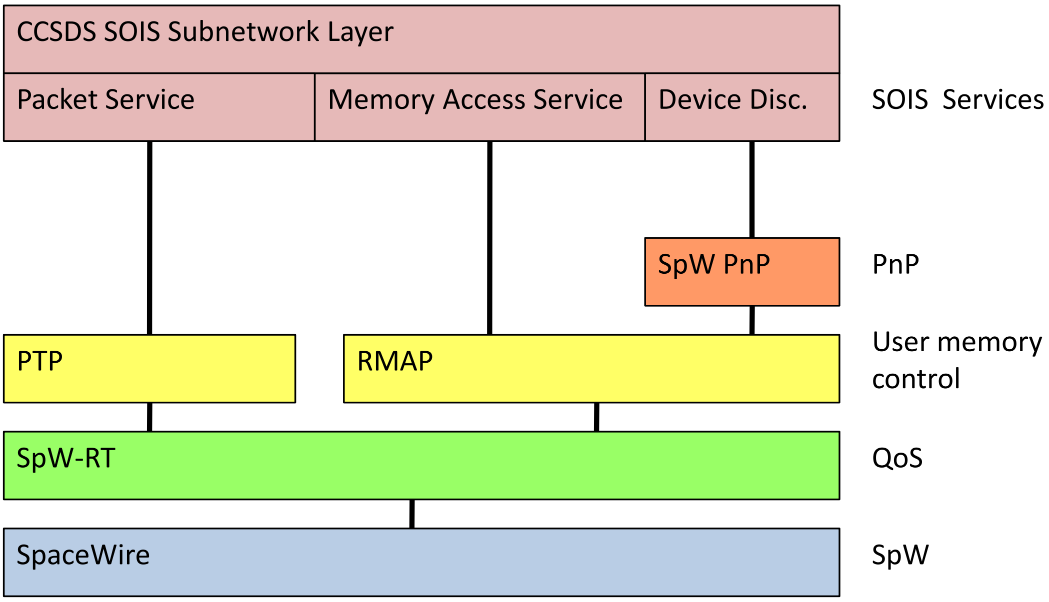


**Figure 2 SOIS to SpaceWire Layered Protocol Stack**

To send a packet across a SpaceWire network using the CCSDS SOIS Packet Service the packet is presented to the SOIS Packet Service and passed to SpaceWire-PTP which manages buffer space for the complete SOIS packet to avoid sending the packet when there is no room for it in the destination. SpaceWire-PTP passes the packet as a stream of characters to SpaceWire-RT which chops the stream into segments, sends them across the network, and reassembles the SOIS packet at the destination. The SOIS packet emerges from SpaceWire-RT as a stream of characters which are read and put into appropriate packet buffer space by SpaceWire-PTP. The complete buffered packet is then made available to the user application via the SOIS Packet Service interface.

SpaceWire protocols are designed for efficient implementation, high performance and both hardware and software implementation. The SOIS services are designed for generic implementation over various buses or networks, providing a standard software interface.

# 4 QUALITY OF SERVICE

SpaceWire-RT provides five quality of service (QoS) classes:

- The **Basic QoS** provides a service which does not ensure delivery (i.e. does not provide any redundancy and does not retry in the event of a failure to deliver) and is not timely (i.e. does not deliver information within specified time constraints).

- The **Best Effort QoS** also provides a service which does not ensure delivery and is not timely. This service is different to the Basic Qos only in that Best Effort does not deliver duplicate or out of sequence packets whereas this is possible with the Basic QoS.

- The **Assured QoS** provides a service which is reliable (i.e. retries in the event of a failure to deliver) but is not timely.

- The **Reserved QoS** provides a service which does not ensure delivery but is timely (i.e. when a packet is delivered it is delivered on time).

- The **Guaranteed QoS** provides a service which is both reliable and timely (i.e. it will retry in the event of a failure to deliver and deliver information on time).

## 5 ASYNCHRONOUS AND SCHEDULED SYSTEMS

To provide timeliness of delivery imposes constraints on the SpaceWire system design. Specifically, a means of ensuring that there is no conflict over the use of network resources is required i.e. every channel has to be given an appropriate share of the available network bandwidth over which it can send information without fear of that communication being held up by other traffic on the network.

Not all systems require timeliness, hence SpaceWire-RT supports two types of system: Asynchronous and Scheduled.

### 5.1 ASYNCHRONOUS

In an asynchronous system the sending of information over the SpaceWire network is asynchronous i.e. SpaceWire packets are sent when there is a packet in the source to be sent and room in the destination to receive the packet. Priority is used to provide a limited mechanism for controlling timeliness of delivery: a high priority packet will be sent before a low priority one, provided that there is room in the high priority destination buffer.

An asynchronous network supports three QoS classes: Basic, Best Effort and Assured.

### 5.2 SCHEDULED

In a scheduled system the network bandwidth is split up using time-codes. Time-codes are distributed across the network periodically. Each time-code indicates the start of a time-slot. Each source channel is assigned one or more time-slots when it is allowed to transmit information. Timeliness of delivery is controlled by a schedule table used to specify which source channel can send information in which time-slot. This provides deterministic delivery.

A scheduled network supports five QoS classes: Basic, Best Effort, Assured, Resource-Reserved and Guaranteed.

## 6 SPACEWIRE-RT SERVICE INTERFACE

To isolate the applications using SpaceWire-RT from the particular type of SpaceWire system being used for communication the same service interface is used for both asynchronous and scheduled systems.

There are six primitives currently defined for the SpaceWire-RT service:

- Send_Data.request (channel, source address, destination address, priority, cargo) which requests to send a Service Data Unit (SDU) from the source node where the request is being made to a destination node on a SpaceWire network;
- Receive_Data.indication (channel, source address, destination address, priority, cargo) which indicates that a SpaceWire-RT packet has been received and which passes the SDU it carried to the SpaceWire user;
- Notify_Delivered.indication (channel, source address, destination address, SDU_ID) which indicates to the user that issued a Send_Data.request over a channel that provided assured or guaranteed services that the SDU was safely delivered to the destination.
- Notify_Error.indication (channel, source address, destination address, SDU_ID, error metadata) which indicates to the user that issued a Send_Data.request that there was a problem delivering the SDU over a channel that provided assured or guaranteed services.
- Configure.request (channel, configuration information) which configures the channel parameters.
- Redundancy_Invocation.indication (channel, reliability metadata) which indicates that one or more retries or redundancy switching were invoked for a channel.

# 7 ARCHITECTURE

To provide the required QoS capabilities the SpaceWire-RT protocol has to implement a number of different functions each of which is described in the following subsections:

## 7.1 SEGMENTATION

User information is passed to SpaceWire-RT for sending across a SpaceWire network. The size of this user information is arbitrary and unknown to SpaceWire-RT. SpaceWire-RT sends information across the SpaceWire network in data protocol data units (DPs) each with a size up to a specific maximum DP size (256 bytes). To fit the user information into one or more DPs it has to be split into segments that fit into the available space in the DPs.

The segmentation function is responsible for splitting up the user information into user data segments no larger than a maximum segment size.

## 7.2 END TO END FLOW CONTROL

End to end flow control manages the source and destination buffers within SpaceWire-RT. A DP for a particular channel cannot be sent unless there is room for it in the destination buffer for that channel. SpaceWire-RT keeps track of the space available in destination buffers for all channels using a flow control mechanism whereby the destination informs the source of the space available in the destination buffer either periodically (scheduled system) or when space becomes available (asynchronous system).

## 7.3 RETRY

To ensure that data is delivered when there are temporary faults on the SpaceWire network a retry mechanism is required. When a DP is sent using the assured or guaranteed QoS the source keeps a copy of the transmitted DP. When the DP arrives at the destination an acknowledgement PDU (ACK) is sent back to the source. When the source receives the ACK it can free the buffer containing the DP. If no ACK is received within a time-out period due to either the DP or the ACK being lost or corrupted, the DP can be resent to recover from a transitory error.

Retry is combined with the redundancy function to give automatic recovery from transitory and permanent faults.

## 7.4 REDUNDANCY

The Redundancy Model adopted by SpaceWire-RT is that of alternative paths from a source node to a destination node across a SpaceWire network. SpaceWire-RT supports autonomous switching between alternative paths. These alternative paths may be used in one of two ways:

- Sending data over both paths at the same time, which is referred to as simultaneous retry.
- Sending over the prime path and then if there is a failure using the redundant path.

## 7.5 ERROR DETECTION

Error detection is needed to support the retry functions and also for error notification for the Basic, Best Effort and Resource Reserved classes of traffic.

## 7.6 ADDRESS TRANSLATION

SpaceWire can provide up to 223 logical addresses, permitting up to 223 separate nodes. This number is adequate for most foreseen space missions, so for node identification SpaceWire-RT uses the SpaceWire logical address. Path addressing may be used to route a packet to its destination but the node identification is done using the logical address.

The address translation function translates from the SpaceWire logical address to the SpaceWire address that will be used to send the packet across the network. The SpaceWire address can be a path, logical, or regional logical address or an address constructed using any combination of these addressing modes. The type of address used is dependent upon the redundancy approach being used. Address resolution is used to determine the SpaceWire address bytes that are included in the header of the SpaceWire packet to route it along the required path across the SpaceWire network to its intended destination.

## 7.7 PDU ENCAPSULATION

The user data segment (UDS), destination address, source address, channel number and other necessary information have to be packaged into a Data PDU (DP) for sending across the SpaceWire network as a SpaceWire packet. The DP encapsulation is illustrated in Figure 3.

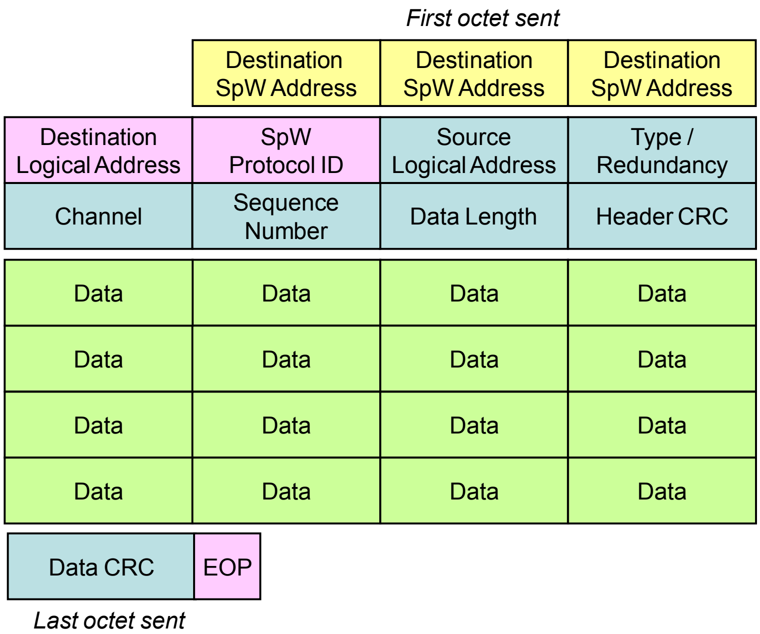


**Figure 3 DP Encapsulation**

Control information, including ACKs, that are necessary for the operation of SpaceWire-RT also need to be encapsulated into control PDUs.

## 7.8 PRIORITY

If there are several DPs waiting to be transmitted, the priority function selects which DP to send next based on its priority, the availability of space in the relevant destination buffers, and, for scheduled systems, the schedule and current time-slot. If there is space in the appropriate destination buffers then the DP with highest priority will be sent first.

## 7.9 RESOURCE RESERVATION

Resource reservation is required for the resource reserved and guaranteed qualities of service to provide timeliness of data delivery. Resource reservation is only provided in the scheduled system which was specifically designed to support it.

In a scheduled network the network bandwidth is separated using time-division multiplexing into a series of repeating time-slots. A schedule table is used in each source to specify which source channel buffer(s) are allowed to send information during each time-slot. The schedule tables in every source are configured to avoid conflicts on the network. Only one source is allowed to send information at a time, or multiple sources can send information at the same time provided that they do not use any common network resource i.e. send information over the same SpaceWire link.

## 8 FUTURE WORK

SpaceWire-RT is defined in substantial detail in the SpaceWire-RT Initial Protocol Definition [4]. An earlier version of this document was made available to the SpaceWire Working Group to indicate progress with the draft standard and the latest version is now available. The document is not yet complete nor at a stage where a complete prototype SpaceWire-RT systems can be designed.

Current effort is focused on filling the gaps in the draft standard, making it consistent with the CCSDS SOIS protocols, and making it ready for review by the SpaceWire Working Group.

## 9 ACKNOWLEDGEMENTS

The authors acknowledge the support of ESA for the present work which is funded by ESA under ESA Contract No. 220774-07-NL/LvH.

## 10 REFERENCES

[1] ECSS, “SpaceWire: Links, nodes, routers and networks”, ECSS-E50-12A, January 2003

[2] ECSS “SpaceWire Protocols”, ECSS-E-ST-50-11C, Draft 1.3, July 2008

[3] S.M. Parkes, “SpaceWire-RT Requirements”, SpaceNet Report No. SpW-RT WP3-100.1, ESA Contract No. 220774-07-NL/LvH, February 2008

[4] S.M. Parkes and A. Ferrer-Florit, “SpaceWire-RT Initial Protocol Definition”, Draft 2.1, SpaceNet Report No. SpW-RT WP3-200.1, ESA Contract No. 220774-07-NL/LvH, October 2008.

[5] CCSDS 850.0-G-R1.1 Draft Green Book Spacecraft Onboard Interface Service Draft Informational Report Jan 2008

[6] CCSDS 851.0-R-1.1 Draft Red Book Spacecraft Onboard Interface Service Subnetwork Packet Service Draft Recommended Practice Jan 2008

[7] CCSDS 852.0-R-1.1 Draft Red Book Spacecraft Onboard Interface Service Subnetwork Memory Access Service Draft Recommended Practice Jan 2008

[8] CCSDS 854.0-R-1.1 Draft Red Book Spacecraft Onboard Interface Service Device Discovery Service Draft Recommended Practice Jan 2008