D2.1 SpaceWire-RT Outline Specification

Lead Beneficiary:	University of Dundee		
Author(s):	Steve Parkes, UNIVDUN		
Work Package:	WP2	**Task:**	Tasks 1, 2, 3, 4, 5, 6, 7, 8
Version:	2.00	**Last modified:**	06/09/2012
Status:	Released		
Approved by:	S.M. Parkes	**Date:**	06/09/2012

Table of Contents

Table of Figures

Table of Tables

1 Introduction

In this section the SpaceWire-RT project is first introduced to give the reader a broad picture of the research and how the work presented in this report fits into the overall objectives of the project. The research covered in work package 2 is then introduced and put in context of the previous and subsequent work packages.

1.1 Introduction to the SpaceWire-RT Project

In this section the SpaceWire-RT project is introduced, starting with the motivation for the proposed research, the project aims, and the background to the proposed research. SpaceWire-RT aims to cover many on board communications applications from low to very high data-rate networks. The key features missing from the existing SpaceWire and SpaceFibre protocols are then considered: Reliability and Timeliness (RT), which are the principal concern of the proposed SpaceWire-RT research programme. The benefits of the proposed programme are then described and the specific research objectives summarised.

Note: this section is taken from the SpaceWire-RT proposal and updated where appropriate.

1.1.1 Motivation

The trend towards "Operationally Responsive Space", where spacecraft can be rapidly assembled, configured and deployed, to meet specific mission needs, e.g. disaster support, requires flexible on board communication networks with plug-and-play capability. The growing autonomy of scientific missions to remote planets requires highly capable on board networks that are robust and durable, able to recover from transitory errors and faults automatically, without complicating the many applications running over the network. The importance of mass reduction on a spacecraft requires the most to be made of on board resources, including the sharing of networks for both payload data-handling and avionics applications. Avionics and robotics impose requirements on network responsiveness and determinism. The increasing international collaboration on scientific and Earth observation spacecraft requires standard network technology where a component developed by one nation will interoperate effectively with equipment developed by another nation. SpaceWire-RT aims to fulfil these demanding requirements with a flexible, robust, responsive, deterministic and durable standard network technology that is able to support both avionics and payload data-handling applications.

SpaceWire [1][2] has proved to be a very successful first step in this direction, providing networking technology for payload data-handling on over 40 major space missions. It falls short, however, of the requirements for avionics systems.

Integration of instruments and equipment from different nations requires alignment in onboard system integration. European and Russian space industry have differences in rules and requirements for system integration (e.g. in galvanic isolation guidelines). The joint development of prospective onboard networking technology in EU/RF collaboration is vital to overcome this problem

1.1.2 Aims of Research Programme

The SpaceWire-RT research programme aims to conceive and create communications network technology, suitable for a wide range of demanding space applications where responsiveness, determinism, robustness and durability are fundamental requirements. This is a critical component technology for future spacecraft avionics and payloads. A quality of service (QoS) layer will be developed for SpaceWire to support mixed avionics and data-handling applications.

SpaceWire-RT will:

- use virtual channel concepts to provide a variety of QoS;
- provide broadcast and multicast capability;
- increase performance;
- provide low latency message delivery;
- include extremely low latency time and out-of band signalling mechanisms;
- incorporate novel fault detection, isolation and recovery methods;
- make the network fully responsible for information transfer;
- decouple application and data transfer;
- implement appropriate communication mechanisms in relatively simple hardware.

The principal focus of the proposed research is spacecraft avionics networks with terrestrial avionics, robotics, automobiles and other applications also expected to benefit from the anticipated technological advancements.

1.1.3 Background

Terrestrial communication networks have become ubiquitous, reaching into most homes and businesses, in cars, aeroplanes, and industrial plants. Ethernet offers ever increasing bandwidth but with limitations in latency and real-time responsiveness. USB offers high bandwidth (especially USB 3.0) and moderate latency but at the expense of a severely constrained topology. PCI-Express, Infiniband and RapidIO are very high bandwidth networks, but are very complex, have topological limitations, and limited FDIR capabilities.

SpaceWire is a data-handling network for spacecraft which combines simple, low-cost implementation, with high performance and architectural flexibility. Its advantages over competing technologies have been demonstrated by its rapid take up by the normally conservative international space agencies and space industry. SpaceWire is now being used on more than 40 high profile missions and by all of the major space agencies and space industry across the world.

SpaceWire is ideal for data-handling applications but does not address avionics and other applications where responsiveness, robustness, determinism and durability are essential requirements. There is a need for a spacecraft avionics network technology which combines the key features of SpaceWire with the quality of service requirements of real-time avionics applications. Mil-Std 1553 has long been the communications bus of choice for spacecraft avionics. Limited to 1 Mbits/s aggregate data rate and constrained to the bus topology, Mil-Std 1553 is struggling to cope with today’s spacecraft requirements. On-board payload data-handling is now dominated by the SpaceWire standard. The need in smaller spacecraft, planetary landers, etc., for integrated avionics and data-handling networks has raised the possibility of using SpaceWire for avionics applications. This requires some fundamental extension to SpaceWire which this research programme aims to address.

1.1.4 SpaceFibre

SpaceFibre [5] is a very high-speed serial data-link being developed by ESA which is intended for use in data-handling networks for high data-rate payloads. SpaceFibre is able to operate over fibre optic and copper cable and support data rates of 2 Gbit/s in the near future and up to 6 Gbit/s long-term. It aims to complement the capabilities of the widely used

SpaceWire onboard networking standard: improving the data rate by a factor of 10, reducing the cable mass by a factor of four and providing galvanic isolation.

SpaceFibre will support high data-rate payloads, for example synthetic aperture radar and hyper-spectral optical instruments. It will provide robust, long distance communications for launcher applications and will support avionics applications with deterministic delivery constraints through the use of virtual channels. SpaceFibre will enable a common onboard infrastructure to be used across many different mission applications resulting in cost reduction and design reusability. SpaceFibre can run over fibre optic or copper cables.

A prototype SpaceFibre interface was designed by the Space Technology Centre at the University of Dundee in 2007, a demonstration system built and an initial draft of a standard document written. A prototype was built to this specification by NASA Goddard Space Flight Centre and flown fly on the MAST test vehicle.

The CODEC design for SpaceFibre has many advantages compared to SpaceWire:

- It uses fewer wires reducing cable mass;
- It operates at data rates of 2 Gbits/s and potentially higher;
- It uses matched impedance connectors;
- The size of all the characters are the same (32-bits);
- Parity coverage is per character;
- It uses a DC balanced encoding scheme;
- It provides simple capacitive, magnetic, or optical galvanic isolation;
- The initialisation protocol is base on a double handshake;

At the time when the SpaceWire-RT project started, SpaceFibre did not provide QoS, and FDIR facilities. These critical areas have been addressed by SpaceWire-RT, and are reported in this document, the results have been fed into the SpaceFibre specification supporting QoS and FDIR capabilities in the latest version of the SpaceFibre specification [6].

SpaceWire-RT aims to build on the development of SpaceFibre technology, enhancing it with appropriate QoS and FDIR mechanisms, and exploring how best to make it backwards compatible with SpaceWire.

1.1.5 Quality of Service: Reliability

Reliability will be provided using three techniques: inherent reliability of the links, retry to recover from transient errors, and redundancy to recover from permanent faults.

Inherent link reliability is achieved through simplicity, appropriate signalling margins, and good EMC control.

Retry can be achieved using an end-to-end or link-by-link protocol. An end-to-end protocol is preferred because it reduces the need for intermediate buffering in the network, and enables more comprehensive fault recovery policies to be implemented. If a link-by-link protocol is used there is, in any case, still a need for an end-to-end protocol.

Redundancy at the network level is achieved by adding extra links and routers to avoid single point failures. The topological flexibility of SpaceWire makes this relatively straightforward.

Substantial work has been done on reliability mechanisms for SpaceWire by University of Dundee as part of the ESA SpaceNet activity [7].

1.1.6 Quality of Service: Timeliness

Virtual channels are proposed for providing timeliness of delivery; both responsive and deterministic. Responsive behaviour requires the ability to send messages with low latency; priority or pre-emption may be used to implement this. Determinism requires reservation of network resources, so that network capacity is available when needed to send information. Determinism may be implemented using time-division multiplexing to split up available network bandwidth, or by allocating, measuring and controlling network resource usage. Virtual channels will be used to support flow of messages with different QoS across links.

An illustration of a virtual channel system is provided in Figure 1-1.

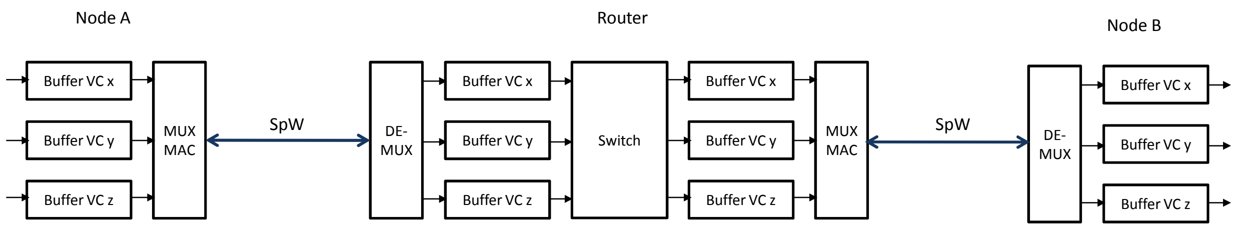

Figure 1-1 Virtual Channels in a SpaceWire System

It shows two nodes connected via a routing switch. Virtual channels (VCs) are shown for one direction of communication only. Data to be sent across the network is placed in the appropriate VC buffer in Node A, depending on the quality of service required. Each VC provides a specific quality of service. The Mux/MAC determines which buffer gets to send information across the SpaceWire link at any moment. This will depend on several things:

- Which VC sending buffers have data to send;
- Which VC receiving buffers have space available to receive data;
- The arbitration or medium access control (MAC) policy in force for each VC.

For SpaceWire-RT several MAC policies will be provided including:

- Priority, where lowest VC number supporting priority goes first;
- Bandwidth reserved, where the VC with allocated bandwidth and recent low utilisation of the link will go first;
- Time-division multiplexed, where time-slots are defined by time-codes and the VC allocated to the current time-slot can send data. If this VC has no data to send then another VC may used this unused bandwidth opportunistically.

Across a SpaceWire link the MAC policy determines which VC can send data. Inside a router VC input buffers are connected to VC output buffers depending on the address of the packet being transferred. VC input buffers are always connected to VC output buffers with the same VC channel number, but the SpaceWire output port that is used depends on the packet destination address. When an end of packet marker is received by a VC output buffer, arbitration will take place inside the router to determine which input port will have access to the output buffer.

The use of VCs and the various medium access policies will enable both responsive and deterministic communications across a SpaceWire 2.0 network.

During the proposed programme of research the literature will be reviewed and appropriate techniques developed to control traffic over the network providing responsiveness and determinism.

1.1.7 Research Objectives

The SpaceWire-RT research programme has the following key objectives:

- A consolidated and justified set of requirements and use cases for SpaceWire-RT covering both spacecraft payload and avionics, which takes into account requirements from spacecraft primes and equipment manufacturers, from EU, RF and internationally.

- A consolidated concept for SpaceWire-RT which takes into account relevant literature, explores and analyses alternative designs, and which trades-off alternative solutions.
- An outline specification for SpaceWire-RT based on the requirements and use cases and the conceptual design and analysis work.
- A validated SpaceWire-RT specification with important, novel features of SpaceWire-RT networks tested using SDL models.
- A validated VHDL IP Core for SpaceWire-RT aimed at FPGA implementation.
- An assessment of the feasibility of implementing SpaceWire-RT as an ASIC core considering available ASIC technologies suitable for space.
- A draft standard document for SpaceWire-RT, which has been reviewed by the International SpaceWire Working Group.
- Disseminated results of the SpaceWire-RT study to the European and Russian space industries, and to the international space community.
- An exploitation plan covering the availability of flight grade chips, test and development equipment, IP cores, and a programme for a test flight.

1.2 Concepts and Specification Development

This section summarises the research presented in this report. First it summarises the preceding research, then an overview of the present research is provided, and finally the next steps are presented.

1.2.1 Preceding Research (WP1)

The requirements and use cases for spacecraft onboard communication networks for both payload applications and avionics were developed in WP1. Requirements were gathered for avionics networks, including reliability, fault tolerance, fault isolation, performance, responsiveness and determinism, based on the experience of SubMicron and EADS Astrium on the many spacecraft they have developed or contributed flight equipment to over the past decades. A series of use cases were provided covering a diverse set of avionics and payload applications. The requirements and use cases were presented to the SpaceWire Working Group and feedback from the group taken into account.

1.2.2 Present Research (WP2)

WP2, the subject of this report, addresses the second and third project objectives:

- A consolidated concept for SpaceWire-RT which takes into account relevant literature, explores and analyses alternative designs, and which trades-off alternative solutions.
- An outline specification for SpaceWire-RT based on the requirements and use cases and the conceptual design and analysis work.

In parallel with the work on the requirements and use cases covered by WP1, relevant literature on real-time network concepts was reviewed (section 2). The requirements of WP1 were analysed (section 3). From this analysis a set of key challenges for SpaceWire-RT were identified (section 4). A set of evaluation criteria were then derived (section 5). SpaceFibre lacked any specification of QoS and a novel, powerful, but simple QoS mechanism was designed and is presented in section 6. FDIR was also considered in section 7, building on and enhancing the capabilities provided by the QoS mechanism, enabling it to monitor and detect various types of system fault. SpaceFibre, including the new QoS and FDIR mechanisms, is introduced in section 8 and reviewed against the requirements from WP1. It is then further evaluated against the derived evaluation criteria in section 9, highlighting where SpaceFibre does not meet the SpaceWire-RT requirements. In section 10 these deficiencies of SpaceFibre are addressed, the missing features are listed and potential solutions to cover these deficiencies are explored. In section 11 a coherent set of protocols is proposed, utilising SpaceFibre for high-speed communication and SpaceWire for lower-speed communication. The QoS and FDIR concepts are applied across the proposed set of protocols. In section an outline specification for SpaceWire-RT is presented, building where appropriate on existing space protocols and enhancing them when necessary.

1.2.3 Subsequent Research (WP3, WP4 and WP5)

With the initial specification available from WP2, work on simulation, IP core development, and ASIC prototyping can now start.

In WP3 simulation models covering key aspects of SpaceWire-RT will be designed and used to evaluate the proposed SpaceWire-RT protocols. WP3 will focus on testing the QoS and FDIR layers of SpaceFibre. The simulation models will be developed in SDL and used to run the various validation scenarios derived from the uses cases. The results of the simulation will be used to update the SpaceWire-RT specification and to inform the VHDL IP Core Development (WP4) and ASIC Feasibility and Prototyping (WP5) activities.

WP4 will begin with the architectural design of key aspects of the SpaceWire-RT IP Core based on information from the outline SpaceWire-RT specification (WP2). The specific aspects to be implemented are “oversampling” and “SpaceFibre over SpaceWire”.

The eventual goal is for SpaceWire-RT technology to be implemented in ASIC devices suitable for spaceflight. Owing to the expense of ASIC devices this is beyond the scope of the present research. However, in WP5 appropriate ASIC technologies will be investigated and initial design and core prototyping activities undertaken, to ensure that the principal risk areas with an ASIC development have been addressed.

1.3 References

[1] European Cooperation for Space Standardization, Standard ECSS-E-50-12A, “SpaceWire, Links, Nodes, Routers and Networks”, Issue 1, European Cooperation for Space Data Standardization, January **2003**.

[2] European Space Agency, “SpaceWire Web Page”, European Space Agency, http://spacewire.esa.int/

[3] Parkes SM et al, “Review of Standard and Status”, Digital Interface Circuit Evaluation Study WP1000 Technical Report, Document No. UoD-DICE-TN-1000, ESA Contract No. 12693/97/NL/FM, University of Dundee, July 1998.

[4] IEEE Computer Society, “IEEE Standard for a High Performance Serial Bus”, IEEE Standard 1394-1995, IEEE, August 1996.

[5] Parkes SM, McClements C, Dunstan M, and Suess M, “SpaceFibre: Gbits/s Link For Use On Board Spacecraft”, International Astronautical Congress, Daejeon, South Korea, October 2009.

[6] Parkes SM, Ferrer Florit A, Gonzalez A, and McClements C, "SpaceFibre", Draft D, Space Technology Centre, University of Dundee, 29th February 2012.

[7] Parkes SM and Ferrer Florit A, "SpaceNet – SpaceWire-RT Protocol Definition", Draft A version 3.2, International SpaceWire Working Group, 27th August 2009.

2 Real-Time Network Concepts Literature Review

In this section a literature review covering real-time network concepts is presented.

2.1 Networking technology features in the SpaceWire-RT project

SpaceWire-RT aims to develop a flexible, robust, responsive, deterministic and durable network technology that is able to support both avionics and payload data-handling applications.

Analysis of space industry demands and requirements shows that SpaceWire-RT should provide a variety of QoS, multicast and broadcast capability; fault detection, isolation and recovery methods, support time synchronization and low latency out-of band signaling. Virtual channels are planned to support. Other networking mechanisms should be analyzed, improved and applied to support the key SpaceWire features.

Development of SpaceWire-RT key features is based on analysis of existing network technologies and their further development with specifics of real-time spacecraft onboard networking. This section gives a review of methods and theoretical approaches for required features support in network technologies and in real-time network communication. A summary of overview and analysis of methods and approaches for Quality of Service, virtual channels, flow control methods, multicasting, network redundancy, fault detection, isolation and recovery is presented in the following subsections.

2.2 Quality of Service

The network traffic may be divided into several classes in order to manage the shared network resources more efficiently. Some traffic classes may have different requirements, may have different levels of priority. Traffic classes fall into two broad categories: **best efforts classes** and **guaranteed service classes**.

The network makes no strong guarantees about best efforts packets. Depending on the network, these packets may be arbitrary delayed or even dropped. The network will simply make its best effort to deliver the packets to their destination [6]. Best effort classes are implemented in TTEthernet [8], Fibre Channel [12] and SpaceFibre [9, 11].

Guaranteed service classes are guaranteed a certain level of performance as long as the traffic complies with a set of restrictions. There is a service contract between the network and the client. As long as the client complies with the restrictions (e.g. the maximum data injection rate), the network will deliver the performance. There are a number of different approaches for guaranteed services implementation, which can be classified by the level of latency deviation.

In a deterministic network the message transmission latency is a function of the message size, the network hardware characteristics (e.g. links data rate) and the message path length (i.e. the number of intermediate switches). On the other hand, incoming to the network data flows do not affect the message transmission latency: a deterministic network is said to provide low level of latency deviation. If a network does not provide that the message transmission latency is not influenced by other data flows, the deviation level of the latency is high and it doesn't meet deterministic network requirements.

The **aggregate resource allocation** approach does not distribute the network resources between particular data flows. Instead of it, this mechanism requires that the aggregate demand of data flows for the resources must be less than a defined bound. This can be implemented by deployment of (σ, ρ) regulator (described in [6]). Thus, the maximum transmission latency guarantee is based on the number of incoming data flows and their injection rates. However, this approach does not provide the low latency deviation and, consequently, low jitter. The cause is the transmission latency of a message depends on other messages transmitted simultaneously through the same switches. On the other hand, this method is not expensive in terms of hardware cost [6]. Such method is used in AFDX protocol.

The second approach is a **resource reservation in space**, which is based on virtual circuit deployment. The method aims to choose an independent route for each virtual circuit in order to distribute the data flows through the network and, consequently, avoid coupling between them. However, as the previous one, this approach does not provide independence between all data flows in the network. Some data flows which are scheduled to use the same links can interfere. Therefore, the approach of resource reservation in space does not guarantee tight latency determinism and low jitter. This approach is used for Fibre Channel.

The third type of resource reservation is **time-division multiplexing** (TDM). This method breaks system time into a number of time-slots allowing one client to capture all the resources. TDM guarantees that the maximum values of latency and latency jitter never exceed correspondent upper bounds and ensure allocated bandwidth for a data flow. It is noticeable that this approach can be used in cases of sharing one resource, e.g. a bus. Method is implemented in TTP/C [7], SpaceFibre, MIL-STD-1553B [13].

When extremely tight guarantees are required, **a combination of the resource reservation in space and time-division multiplexing** provides the strictest controls. This method locks a piece of the resources for a particular flow both in space and in time. Because only one flow uses these resources, its latency and jitter deviations are as minimal as possible. In order to implement this, the system operation time is divided in a number of time-slots and a schedule defines when a certain data flow is allowed to use the resource. In general, finding an optimal schedule is NP-hard, so most practical implementations resort to heuristic approaches [6]. A disadvantage of this method is that strict configuration does not allow the addition of new nodes or messages without redesigning the message and task schedule [1]. An example of this method’s applicability is TTEthernet.

Implementation of a selected QoS methodology requires support of other network mechanism to provide strict segmentation and partitioning of shared network resources in accordance with QoS algorithm to ensure required real-time traffic indexes irrespective to any traffic sources behavior. Virtual channels are prospective concept for its implementation.

2.3 Virtual Channel Concept

The virtual channel concept is a widely used mechanism in order to separate data flows in a network and provide some form of data flows independence while sharing network infrastructure resources. There are three distinct concepts presented in literature and practice: Point-to-Point (Virtual Channel), End-to-End (Virtual Circuit) and Virtual Networks.

The first one is a point-to-point **virtual channel** concept. That is, several unidirectional virtual channels are multiplexed across a physical link [3]. Each virtual channel independently manages two data buffers: one in the transmitter and one in the receiver. The data flow through the virtual channel is shaped by the independent flow control mechanism. Logically, each virtual channel operates as if there is only one data flow over the physical link. The virtual channels may share the bandwidth of the physical link on the flit-by-flit or frame-by-frame basis or any other fair scheme. This mechanism has been intended for deadlock avoidance. Because data flows of different virtual channels do not affect each other and the resource separation is fair, no message can occupy the link and, consequently, block other transmitted data. The more separate virtual channels the less possibility of all channels are blocked. This approach is used in SpaceFibre standard.

End-to-end **virtual circuits** are established as one-to-one or one-to-many connections. Each virtual circuit has output buffer in the source, input buffer in the destination(s), flow control and acknowledgement mechanisms (optional) and attributes (e.g. priority level, type of service provided, etc.). A virtual circuit can be established statically or dynamically and does not capture hardware resources (e.g. buffers) in intermediate switches. This concept can be used for routing or resource reservation for data flows and an example of it is AFDX [5].

A **virtual network** is a subset of virtual circuits, which provides end-to-end interconnection of nodes connected to this virtual network. The channel sets corresponding to different virtual networks are disjoint. Consequently, in difference from the virtual circuit concept for each virtual network buffers in intermediate switches are allocated. Thus, due to the high hardware overhead only few virtual networks may be implemented in a network. Virtual network concept is used to make that different traffic classes do not affect each other and, consequently, to avoid deadlocks and guarantee performance.

2.4 Multicast implementation approaches

A multicast can be realized either by message replication before routing (MRBR) or by message replication while routing (MRWR) [14, 15, 16]. In MRBR [17], a multicast message that is destined to *N* nodes is copied *N* times before it is sent into the network. Then, those *N* messages are processed as unicast messages and each of them sent to one of the required destinations. In MRWR [18], the multicast message is sent as one message into the network. It proceeds through the network until it reaches a switch from which no single way leads to all required destinations. There, the message is copied as many times as there are different ways to reach all desired destinations. Such switches may be passed several times [14]. As an example, multicasting is implemented in AFDX [5], Fibre Channel [12] and TTEthernet [8] protocols.

2.5 Basic flow control methods

Flow control management in a network is based on two basic methods: credit-based method and on/off method, [6]. Both types of management provide backpressure by informing the transmitting side whether it is permitted to send data.

There are two general principles of **credit-based flow control**. The first one engages "absolute" credits, i.e. the communicating sides count how many data has been transmitted since the establishment of the connection. This is deployed, in particularly, in InfiniBand [19] and UniPro [20] standards. The second uses more traditional incrementing/decrementing scheme while the transmitting side is informed about the amount of free room in the receiver at the moment. Such an approach is used in SpaceWire [10], SpaceFibre [9, 11] and Fibre Channel [12].

Absolute credit-based flow control methods assume the following principles. The receiving side counts the amount of free space in its buffer (e.g. in bytes). When data is loaded into the buffer the free space counter is not decremented while when data is extracted from the buffer the free space counter is incremented. It is important that the counter shall always be incremented with modulo *N* in order to prevent its overflowing where *N* is a quite big number. The receiving side shall periodically indicate its free space counter value to the transmitting side. This, in turn, shall count the amount of sent data (e.g, in bytes) and increment it in the same way. Thus, the transmitting side is permitted to send more data if the following condition is satisfied:

$$Message\_Length \geq Free\_Space - Sent\_Data \,,$$

where the *Message_Length* parameter value is a length of message (e.g. in bytes) that is ready to be transmitted.

The **relative credit-based flow control** includes the following mechanism. When a receiving side has a room for one more flit to receive it shall transmit a credit to the transmitting side and reserve this room. Upon the reception of the credit the transmitting side increments by 1 its flow control credit counter. Thus, it is permitted to send one more flit. Each time when the transmitting side sends a flit it shall decrement by 1 its flow control credit counter. If the flow control credit counter reaches zero, it means that reception buffer of the receiving side is full and the transmitting side is prohibited to send data. It is possible to credit more than one flit. That is, the receiving side sends credits when a room for N flits is free causing the transmitting side to increment its flow control credit counter by N.

With the **on/off flow control** method the receiving side of a connection informs the transmitting side whether it is permitted to send data or not. Thus, the transmitting side may send out flits since it has been permitted and until it is prohibited to send. No credits are counted.

2.6 Fault Detection, Isolation and Recovery

Real-time networks for spacecraft avionics require prevention, detection and localization of failures in their operation.

There are two classes of faults at the system level: a node failure and a link failure. The *node failure* means that the entire end system or switch fails. In turn, the *link failure* refers to the failure in any communication link. Each link or node failure leads to appearance of a *fault region*. These fault regions should not lead to disconnection of the network.

Faults in a network can be *permanent* or *transient*. Permanent failures remain in the network until it is repaired. There are two types of permanent failures: *static* and *dynamic*. The first type of failures is already in the network as it powered on. The second type of failures occurs randomly during the operation of the network. Transient fault remains in a network for some period of time, after which the network operation is recovered [3].

There are several approaches for detection of failures in the network. The first one is a **k-neighborhood diagnosis.** According to this approach each node in a network records the status of all faulty nodes within distance *k*. The diagnosis of faults can be performed by different means, which are out of scope of this clause. However, some nodes may be unreachable due to faults in components. Therefore, it may be impossible to collect information about all nodes within specified distance. In order to solve this problem, the less restrictive *k-reachability* diagnosis is used, according to which a non-faulty node can determine the status of each faulty node within distance *k* that is reachable via non-faulty nodes [3, 21, 22].

The other way of failure detection is **membership service** use. To tolerate the failure of a node, nodes are grouped into fault-tolerant units (FTU). The purpose of an FTU is to tolerate the failure of a single node inside the FTU. It is essential that the failures are reported to a diagnostic system so that the failed node can be repaired [2].

The failure of an FTU must be reported to all operating FTU with a low latency. This can be achieved by the *membership service*. The membership of a component can be established at a point in real-time which is called a *membership point*. In order to address the membership service, the time-division multiplexing shall be implemented. In this case every receiver knows a priori when a message of a sender is supposed to arrive. According to this fact, the arrival of the expected message indicates that the sender node operates correctly. Otherwise, the sender is considered to be failed [2].

In order to distribute status information among all FTUs in the cluster, they can exchange their membership information. By means of this exchange FTUs will be able to correct invalid membership fields.

2.7 Redundancy in a network

Most efficient way to satisfy fault-tolerance requirements is to provide redundancy in the network, so that the network can tolerate a certain number of faults [3]. In order to tolerate all types of hardware faults the redundancy approach can be used.

A link is said to be *redundant* if, after removing it, the resulting network still provides all required connections. Fault-tolerant network should be designed in such a way that all the channels are redundant, thus avoiding a single point of failure. However, it is not guaranteed, that the network will tolerate several simultaneous faults [3].

A network is said to be *f fault tolerant* if any *f* components are redundant. The *f fault recoverable* network guarantees that for any *f* failed components in the network, the undelivered message must not hold network resources and induce deadlock [3]. For example, AFDX, TTP/C and TTEthernet are one fault tolerant.

There are three general kinds of redundancy: hardware, information and time [4]. Each of them is intended for particular situations and has both advantages and drawbacks.

The first type, **hardware redundancy**, is provided by incorporating extra hardware into the design to override the effects of a failed component [4]. For example, two communication nodes (source and destination correspondently) are connected by N communication channels, where $N > 1$. When the source sends out a message to the destination, a replica of the message is sent to each of the communication channels. In the destination, if at least one correct instance of the message is received, the transaction is assumed to be correct. It is clearly seen, that if all interconnections use at least N communication channels, the network is N-1 fault-tolerant. This method is **static hardware redundancy** [4], which is used to tolerate any N-1 faults. Its advantage is that a fault occurrence does not affect the performance of the system. On the other hand, there is **dynamic hardware redundancy**, where shadow components are activated upon a failure [4]. The **combination** of static and dynamic types is possible. Hardware redundancy leads to significant (N times) overhead, which restricts its deployment by safety critical systems.

Information redundancy is implemented by extra bits, which are added to transmitted data (e.g. checksum). If data were corrupted during the transmission, the receiving side may either detect (in case of error-detecting code use) or even recover it (if an error-correcting code is used). This type of redundancy may be used to protect data communicated over noisy channels, which are subject to many transient failures. A disadvantage of the approach is that the additional data consume channel bandwidth.

For networks that are not fault-tolerant **time redundancy** can be used, i.e. a lost message is retransmitted [2]. This approach is effective to tolerate transient failures. The necessity of a message retransmission may be caused by two events. The first is generated by the destination, which has detected the message corruption on the basis of information redundancy. The second is generated by source, when the expected acknowledge on the message has not been received within time interval. The advantage of the approach is relatively low hardware overhead. However, the use of time redundancy increases the latency jitter significantly [2] (it could be unacceptable for real-time data traffic)c and decreases network performance [4].

2.8 Clock Synchronization

Another problem for the real-time data transmission is **clock synchronization**. It deals with the problem that internal clocks of several nodes may differ after some amount of time due to clock drift, caused by clocks counting time at slightly different rates. There are several problems that occur as a repercussion of clock rate differences and several solutions.

The clock synchronization algorithm is given in [2]. Internal clocks of the network nodes shall be periodically resynchronized to establish a global time base with a specified precision Π. The period between two consecutive synchronization events is called as the resynchronization interval R_{int}. The convergence function Φ denotes the offset of the time values immediately after the resynchronization. After resynchronization the clocks drift again and the drift offset Γ defines the maximum possible divergence of any two good clocks from each other during the resynchronization interval R_{int}. To evaluate Γ it is necessary to define the maximum clock drift rate p, which is accepted in the network. The relationship between the resynchronization interval duration, the drift offset and the drift rate is determined by the formula (1):

$$\Gamma = 2pR_{\text{int}} \qquad (1)$$

On the other hand, resynchronization interval R_{int} should exceed significantly the time which is required by the network to recover from a failure of the synchronization message distribution mechanism.

The convergence function, the drift offset and the precision are correlated as following:

$$\Phi + \Gamma \leq \Pi \quad (2)$$

This condition claims that the synchronization algorithm must bring the clocks so close together that during the next resynchronization interval the devices divergence will not exceed the precision interval. There are a number of widely used algorithms that could be applied for the clock synchronization.

The **central master synchronization** is a simple non-fault tolerant synchronization algorithm. It is implemented in TTEthernet and Fibre Channel standards The central master node periodically sends the value of its time counter in synchronization messages (or specific time-codes, as in the SpaceWire [10]) to all other nodes, which are the synchronization slave nodes. Taken into account the known latency of the synchronization message transmission, a slave node calculates the difference between its local clock and the master's time, which is contained in the message. The slave then corrects its clock by this deviation.

The convergence function Φ of the central master synchronization is determined by the *distribution jitter* ε. This jitter is estimated by the following rule. Assume that there is a network with one Central Master node and *N* slave nodes. Let T_i to be the time of transmission of a message/code from the Central Master to the *i-th* slave. Because the distance between Central Master and a slave may vary for different slaves, the T_i values may vary too. Therefore, the distribution jitter is calculated as the difference between the highest and lowest T_i values:

$$\varepsilon = \max(T_i) - \min(T_i), \textit{ where } i = 1,2,..., N \quad (3)$$

The precision of the algorithm is given by:

$$\Pi_{Central} = \varepsilon + \Gamma \quad (4)$$

The central master synchronization is simple, but not fault tolerant, since a failure of the master ends the resynchronization. In a multi-master variant of the algorithm if the active master fails silently and the failure is detected by a shadow master, this shadow master assumes the role of the master and continues the resynchronization.

Such kind of clock synchronization is provided by a **Fault-Tolerant-Average** (FTA) algorithm. It consists of three phases:

1. All nodes which are responsible for clock synchronization exchange their global time counter values.
2. Every node analyzes the collected information to detect errors and executes the convergence function to calculate a correction value for the global time counter.
3. The global time counter is corrected by the evaluated value.

FTA algorithm can tolerate *k* Byzantine faults (described in [2]) in a system of *N* nodes. It is a one-round algorithm that works with inconsistent information and bounds the error introduced by the inconsistency. When a node has received global time values from the other nodes in the system, it shall calculate differences between its global time value and received values. After it the differences are sorted by size and *k* largest and *k* smallest ones are removed. The remaining N-2k time differences are assumed to be within the precision window. The average of these remaining time differences is the correction term for the node's clock. If *k* out of *N* nodes in a system clocks behave in a Byzantine manner, the precision of FTA synchronization can be evaluated by the formula:

$$\prod(N,k,\varepsilon,\Gamma) = (\varepsilon+\Gamma)\frac{N-2k}{N-3k}, \qquad (5)$$

where ε is a latency jitter and Γ is a drift offset determined by the duality of the selected oscillator and the length of the resynchronization interval.

External synchronization links the global time of a cluster to an external standard of time. Such system requires a time server, i.e. a source of an external time, and a time gateway, which is an interface node between the time server and the cluster. The time message, generated periodically by the time server, raises a synchronization event in the cluster and must identify this synchronization event on the agreed time scale. In a fault-tolerant system, the time-gateway should be a fault-tolerant.

Every clock synchronization algorithm demands a clock correction in a local node. There are two ways to do this. The first is so-called **state correction**, when a correction term shall be applied to the clock immediately after resynchronization. This approach is simple to implement, but the divergence may appear each synchronization cycle. On the other hand, the second approach, called as **rate correction**, assumes to modify the rate of the clock so as to speed up or slow down the clock during the next resynchronization interval and, consequently, to bring the clock into better agreement with the rest of the ensemble [2].

2.9 References

1. D.A. Gwaltney and J.M. Briscoe, "Comparison of Communication Architectures for Spacecraft Modular Avionics Systems", Marshall Space Flight Center, Alabama, June 2006.
2. H. Kopetz, "Real-Time Systems. Design Principles for Distributed Embedded Applications. Second Edition", Springer, 2011.
3. J. Duato, S. Yalamanchili, L. Ni, "Interconnection Networks. An Engineering Approach", Elsevier Science, 2003.
4. Israel Koren and C. Mani Krishna, "Fault Tolerant Systems", Elsevier Science, 2007.
5. Aircraft Data Network Part 7 Avionics Full-Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664, Part 7, Aeronautical Radio, Inc., June 27, 2005.
6. W.J. Dally and B. Towles, "Principles and Practices of Interconnection Networks", Elsevier, Inc., 2004.
7. H. Kopetz and G. Grunsteidl, "TTP – A Time-Triggered Protocol for Fault-Tolerant Real-Time Systems", available on-line at: http://www.control.aau.dk/~ppm/P7/distsys/kopetz_ftcs93_TTP.pdf
8. TTEthernet Specification, TTTech Computertechnik AG, November 20, 2008, may be issued at: http://www.ttagroup.org/ttethernet/specification.htm
9. S.M. Parkes, C. McClements and M. Dunstan, "SpaceFibre Outline Specification", University of Dundee, October 31, 2007.
10. ECSS Standard ECSS-E-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", European Cooperation for Data Standardization, July, 2008.
11. S.M. Parkes, "Virtual Channels, Broadcast Channels, SpaceFibre and SpaceWire 2.0", SpaceWire Fourth International Conference, 2011.
12. Fibre Channel Avionics Environment Rev 2.3, INCITS working draft proposed Technical Report, American National Standards Institute, Inc., February 7, 2002.

13. Interface Standard for Digital Time Division Command/Response Multiplex Data Bus
14. D. Tutsch, “Performance Analysis of Network Architectures”, Berlin, Springer, 2006.
15. M.-H. Guo and R.-S. Chang, “Multicast ATM switches: Survey and performance evaluation”, ACM Sigcomm: Computer Communication Review, 28(2):98–131, April 1998.
16. Y. Xiong and L. Mason, “Analysis of multicast ATM switching networks using CRWR scheme”, Computer Networks and ISDN Systems, 30:835–854, 1998.
17. H. Xu, Y. Gui and L.M. Ni, “Optimal software multicast in wormhole-routed multistage networks”, IEEE Transactions on Parallel and Distributed Systems, 8(6):597–606, June 1997.
18. S. Bhattacharya, G. Elsesser, W.-T. Tsai, and D.-Z. Du, “Multicasting in generalized multistage interconnection networks”, Journal of Parallel and Distributed Computing, 22(7):80–95, July 1994.
19. InfiniBandSM Trade Association, “InfiniBandTM Architecture Specification Volume 1”, Release 1.2.1, November 2007.
20. Mobile Industry Processor Interface Alliance. Unified Protocol Version 1.1. 2010. http://en.wikipedia.org/wiki/UniPro#Versions_and_roadmap
21. D. Blough and S. Najand, “Fault tolerant multiprocessor system routing using incomplete diagnostic information,” Proceedings of the 6th International Parallel Processing Symposium, pp. 398–402, April 1992.
22. D. Blough and H. Wang, “Cooperative diagnosis and routing in fault-tolerant multiprocessor systems,” Journal of Parallel and Distributed Computing, vol. 27, pp. 205–211, June 1995.

3 Requirements Analysis

This section provides the results of the requirements analysis. First the requirements from D1.1 Consolidated Set of Requirements for SpaceWire-RT are listed. The key issues from the requirements are highlighted including fault detection, isolation and recovery and quality of service. These key issues are then examined in detail.

3.1 Requirements Summary

The requirements from D1.1 Consolidated Set of Requirements for SpaceWire-RT are listed below. Note the rational for each of the requirements can be found in D1.1. The comments provide an initial analysis of the requirements.

Table 3-1: Requirements Summary

Req#	Title	Requirement	Comment
10	Data Rate (data handling)	Shall be capable of data rates up to 20 Gbits/s.	For very high data rate instruments.
11	Data Rate (all others)	Shall be capable of data rates up to 400 Mbits/s.	For moderate rate instruments and all other types of application (e.g. control).
20	Distance (control bus)	Shall operate over a distance of up to 100 m.	For the largest possible spacecraft and launcher applications.
21	Distance (all others)	Shall operate over a distance of 1 m to 10 m.	For normal sized spacecraft and most data handling applications.
30	Galvanic isolation (control bus)	Shall provide galvanic isolation for long distant applications.	Essential for long distances where there can be a significant common mode potential between the two ends of the line.
31	Galvanic isolation (all others)	Should provide galvanic isolation for in-box applications.	This comment is a “May” in D1.1. It has been changed to a “should” as galvanic isolation can help to prevent fault propagation.
40	Transmission medium	Shall operate on twisted pair, co-ax or	The media shall also include

	(data handling, control bus, computer bus)	optical fiber.	PCB back planes. Different media are mainly for different applications/distances.
41	Transmission medium (telemetry bus)	Shall operate on twisted pair.	This requirement is in effect redundant as it is covered by REQ40.
50	packet size (data handling, computer bus)	Shall support application packet sizes up to at least 32 Mbytes.	This is the largest known SpaceWire packet sized used in an existing spacecraft.
51	packet size (control bus, telemetry)	Shall support packet sizes in the range from 8 bytes to 64 Kbytes.	Smaller packet sizes are sufficient for control and housekeeping telemetry applications.
60	Maximum latency (control bus)	Shall support a maximum latency of less than 100 µs.	To be able to support control loops running at 1kHz.
61	Maximum latency (time synchronization bus)	Shall support a maximum latency of up to 100 ns.	This is a very stringent requirement and will be difficult to meet. A relaxation of this requirement to 1 µs (TBC) would make it more achievable. It might be necessary to provide a separate time signal for those applications that require better than 1 µs (TBC) time accuracy.
62	Maximum latency (computer bus)	Should support a maximum latency of less than 100 ns over a single link.	Again this is a difficult requirement to meet, but it is not a mandatory requirement.
70	Reliability	Shall provide a capability for reliable data delivery.	This is essential for many scientific applications where loss of data is not acceptable.
80	Determinism	Shall provide determinism.	Essential for control applications like GNC. Determinism means the ability to deliver data within specific minimum and maximum time constraints, i.e. at a particular time +/- some tolerance.
90	Validity	Shall support a message bit error rate of less than 10-15.	It is assumed that the BER is 10-12 for the basic signal wire and

			that some form of error detection and recovery is required to achieve higher BER. With a spacecraft handling multiple Gbits/s of data (e.g. 10 Gbit/s aggregate data rate), a 10-12 BER corresponds to one error every 100 s. Improving the BER to 10-15 results in one error every day. This means that an automatic retry mechanism is essential to avoid continuously invoking application software to resolve temporary errors on the line.
100	Automatic acknowledgement (control bus)	Should support configurable automatic acknowledgement.	This is an end to end acknowledgement. SpaceFibre provides a link by link acknowledgement of frames.
110	Automatic fault detection	Shall support automatic fault detection.	Essential for recovery from transient errors.
111	Automatic fault identification	Should support automatic fault identification.	This requirement is provided as a "May" in D1.1 but ought to be a "Should". I.e. a desirable requirement rather than an exception to another requirement.
120	Failure and fault tolerance of network (data handling network)	Network should be able to automatically recover from faults.	Since it is at the network level this implies automatic reconfiguration of a network in the event of a permanent fault, rather than retry of failed communication over a link.
130	Multi-path transmission (control bus)	Shall support multi-path transmission.	This is a robust mechanism to improve the robustness of communication.
140	broadcast data transfer (time synchronisation bus)	Shall support broadcast data transfer.	Necessary to distribute time and synchronization information.

150	multi-cast data transfer (computer bus)	Shall support multi-cast data transfer.	To deliver the same data to the devices in the redundant system.
160	out-of-band signals	Shall transfer synchronization signals and interrupts with very short latency.	To replace single wires used for distributing the time ticks and interrupts.
170	mass interconnect	Shall be less than 30 g/m (for one lane).	This is a target for cable manufacturers and ought to be a "should" i.e. a desirable requirement.
180	power consumption	Should be less than 200 mW	In D1.1 this is specified as a range 50 mW to 200 mW.
190	Communication	Shall support the communication requirements as described in Table 3-2.	

Table 3-2 provides a qualitative summary of the communication requirements for SpaceWire-RT considering critical characteristics against application area.

Table 3-2 Qualitative Communication Requirements

	Distance	Rate	Latency	Packet size	QoS
Data-handling network	Short to long	Low to very high	Not important	Short to long	Reserved bandwidth
Control bus	Short to long	Low	Low	Short to long	Deterministic delivery
Telemetry bus	Short to long	Low	Low	Short	Reserved bandwidth
Computer bus	Short	Very high	Low	Short to long	Reserved bandwidth
Time-sync bus	Short to long	Low	Very low	Short	High priority
Side-band	Short	Low to high	Very low	Short	High priority

These requirements are qualitative and have been translated into verifiable requirements in the following table.

Req#	Title	Requirement	Comment
200	Data-Handling Distance	Shall support communication over distances of a few cm to 100m.	Different communication are necessary for the different lengths e.g. PCB track for short distances and fibre optics for the long distances.
201	Data-Handling Rate	Shall provide communication data rates of 1 Mbits/ to 20 Gbits/s.	Different speeds of communication will be supported by different physical and encoding techniques.
202	Data-Handling Latency	Shall support a maximum end to end latency of less than 100 µs over a distance of 100 m and through 10 routing switches.	Latency is not important for data-handling applications.
203	Data-Handling Packet Size	Shall support packet sizes ranging from 8 bytes to 32 Mbytes.	The size of the packet depends on the end user application. SpW-RT has to support all data handling applications.
204	Data-Handling QoS	Shall provide reserved bandwidth QoS.	A typical instrument wants to send data to a mass memory unit over a point-to-point link or equivalently a virtual point-to-point link with reserved bandwidth.
210	Control Bus Distance	Shall support communication over distances of a few cm to 100m.	Short distances are required for chip-to-chip communication. 100m is required for launcher applications.
211	Control Bus Rate	Shall provide communication data rates of up to 10 Mbits/s.	The data rate requirement is low for control applications.
212	Control Bus Latency	Shall support a maximum end to end latency of less than 100 µs over a distance of 100 m and through 10 routing switches.	To be able to support control loops running at 1kHz. Note there is a discrepancy between the requirement in Table

			3-1 and that in Table 3-2.
213	Control Bus Packet Size	Shall support packet sizes ranging from 8 bytes to 64 kbytes.	Control bus communication does not require large packet sizes.
214	Control Bus QoS	Shall provide deterministic QoS.	Deterministic data delivery is essential for control loop applications.
220	Telemetry (HK) Distance	Shall support communication over distances of a few cm to 100m.	Short distances are required for chip-to-chip communication. 100m is required for launcher applications.
221	Telemetry (HK) Rate	Shall provide communication data rates of up to 10 Mbits/s.	The data rate requirement is low for telemetry applications.
222	Telemetry (HK) Latency	Shall support a maximum end to end latency of less than 100 μs over a distance of 100 m and through 10 routing switches.	Latency is not important for housekeeping applications.
223	Telemetry (HK) Packet Size	Shall support packet sizes ranging from 8 bytes to 1 Mbytes.	Housekeeping communication does not require large packet sizes, but a larger packet size might be appropriate for software downloads etc.
224	Telemetry (HK) QoS	Shall provide reserved bandwidth QoS.	Housekeeping telemetry requires a virtual network from the unit gathering housekeeping information to all the units that are providing the data. A small amount of network bandwidth needs to be reserved for housekeeping.
230	Computer Bus Distance	Shall support communication over distances of a few cm to 1m.	A few cm covers chip-to-chip communication. 1 m covers communication between boards in a box across a backplane and also between boxes close to one another.
231	Computer Bus Rate	Shall provide communication data rates of up to 20 Gbits/s.	High speed communication is vital for computer bus applications.

232	Computer Bus Latency	Shall support a maximum link latency of less than 100 ns over a distance of 1 m.	Latency is critical for computer applications and one of the driving requirements that leads to multi-lane operation. This is a very difficult requirement and might need to be relaxed.
233	Computer Bus Packet Size	Shall support packet sizes ranging from 8 bytes to 32 Mbytes.	The computer bus communication will depend on the size of the data structure being transferred.
234	Computer Bus QoS	Shall provide reserved bandwidth QoS.	Computer bus applications require a virtual network from a processor to all the other processor, memory units and I/O units that it wants to send or receive data from. This virtual network would typically have significant bandwidth allocated to it. For a dependent multi-processor system, deterministic communication might be more appropriate.
240	Time-Sync Bus Distance	Shall support communication over distances of a few cm to 100m.	Time synchronisation requires a time-sync bus running from the time-master(s) to all the nodes that require time synchronisation.
241	Time-Sync Bus Rate		Data rate is not an important requirement for time synchronisation. The data rate is very low, but the latency has to be very low.
242	Time-Sync Bus Latency	Shall support a maximum link latency of less than 100 ns over a distance of 10 m.	Latency is the critical requirement for time synchronisation. A latency of 100 ns over a 10 m link is a demanding requirement. A latency of 1 µ is a more realistic requirement (TBC).
243	Time-Sync Bus Jitter	Shall have link latency jitter of less than 100 ns.	Jitter is a more important requirement than latency, since the nominal latency can be

			calibrated out by a system, although this will require special circuitry.
244	Time-Sync Bus Data Size	Shall support transfer of up to 8 bytes of time information.	Time-sync signalling will not use normal packet transfer mechanisms. 8 bytes is sufficient to hold CCSDS unsegmented time information.
245	Time-Sync Bus QoS	Shall provide high priority QoS.	The key issue is low latency communication. Time information has to have higher priority that other information being sent over the link.
250	Sideband Bus Distance	Shall support communication over distances of a few cm to 100m.	Table 3-2 suggests short distances are required for sideband signalling, but this is equally important for launcher applications.
251	Sideband Bus Rate		Data rate is not an important requirement for sideband signalling. The data rate is expected to be very low, but the latency has to be low.
252	Sideband Bus Latency	Shall support a maximum end to end latency of less than 10 µs over 100 m and across a network comprising 10 routers i.e. less than 1 µs per link including distribution by a router.	End to end latency is the critical requirement for time synchronisation. A latency of 10 µs is appropriate for supporting error or interrupt signalling over a large network.
253	Sideband Bus Packet Size	Shall support transfer of up to 8 bytes of error or interrupt information.	Sideband signalling will not use normal packet transfer mechanisms. 8 bytes is sufficient to hold information about the source and nature of the signal.
254	Sideband Bus QoS	Shall provide high priority QoS.	The key issue is low latency communication. Sideband signalling has to have higher priority that other information being sent over the link, except for time-synchronisation

			information.

Note that some of the requirements in Table 3-2 are duplicates of those in Table 3-1, but they have been kept to simplify traceability from the original set of requirements.

4 Key Challenges for SpaceWire-RT

In this section the key challenges for SpaceWire-RT derived from the requirements analysis and the SpaceWire-RT proposal, are considered. Each key challenge is described and used to define the evaluation criteria in section 5.

First the limitations of existing SpaceWire technology are considered, and then the need for a coherent set of network protocols covering the full range of spacecraft applications is described. FDIR and QoS are important features missing from SpaceWire and are explored. Finally the possibility of backwards compatibility with SpaceWire is investigated.

4.1 Limitations of SpaceWire

SpaceWire is a data-handling network for spacecraft which combines simple, low-cost implementation, with high performance and architectural flexibility. Its advantages over competing technologies have been demonstrated by its rapid take up by the normally conservative international space agencies and space industry. SpaceWire is now being used on more than 30 high profile missions and by all of the major space agencies and space industry across the world.

SpaceWire is ideal for spacecraft data-handling applications but does not address avionics and other applications. There is a need for a spacecraft avionics network technology which combines the key features of SpaceWire with the quality of service requirements of real-time avionics applications. Mil-Std 1553 has long been the communications bus of choice for spacecraft avionics. Limited to 1 Mbits/s aggregate data rate and constrained to the bus topology, Mil-Std 1553 is struggling to cope with today's spacecraft requirements. The need in smaller spacecraft, planetary landers, etc., for integrated avionics and data-handling networks has raised the possibility of using SpaceWire for avionics applications. This requires some fundamental extension to SpaceWire which SpaceWire-RT aims to address.

When looking at how to extend and enhance SpaceWire it is important to consider the current limitations of SpaceWire:

Cable Mass: The data encoding and signalling technique used in SpaceWire is easy to implement in any FPGA or ASIC technology but requires four twisted pairs in the cable. This results in relatively high mass cables.

Data Rate: The speed of transmission is limited to around 200 Mbits/s using current space qualified FPGA technology. Many emerging applications require data-rates of more than ten times this rate.

Matched Impedance Connectors: The 9-pin micro-miniature D-type connectors specified for SpaceWire operate successfully up to 200 Mbits/s and are readily available in flight qualified form. However, they are not controlled impedance. A matched impedance connector is essential at speeds above 200 Mbits/s.

Character Sizes: SpaceWire uses four different length codes: control 4-bit, NULL 8-bit, data 10-bit and time-code 14-bit). Handling these different sized characters complicates the transmitter and receiver circuitry.

Parity Coverage: The parity bit covers the data/control field from the previous character and the data/control flag from the next character. This approach complicates both the transmitter and receiver because two characters have to be considered together to determine the parity value.

Transmitted DC Component: SpaceWire characters use all possible bit patterns of the 10-bit data and 4-bit control characters. Depending on the data pattern sent there will be a DC component to the transmitted signal. This prevents AC coupling from being used.

Initialisation Protocol: The initialisation protocol in SpaceWire is based on part handshake and part timing. This can lead to false initialisation caused by noise and data characters being sent due to noise, before a parity error or other error is detected and the link is terminated.

Galvanic Isolation: Because of the DC component SpaceWire does not provide a method of galvanic isolation. A technique using capacitive coupling and bus-hold circuits has been proposed for SpaceWire [3] as used in IEEE1394 [4], but this requires additional isolated power supplies.

Packet Blocking: When SpaceWire routers are being used, it is possible to have two (or more) packets from different sources that have to travel over a particular SpaceWire link to reach the intended destination. If one of the packets is travelling over this link, the second one has to wait until the first one has finished: it is blocked waiting for the link to become free. If the first packet is a large one, the second packet might have to wait a long time before it can progress across the network. In the meantime the blocked might be blocking other packets from moving across the network, because it might be strung out across several routers and links stretching back to its source.

Transport Functionality: SpaceWire lacks transport layer functionality, so there is no consistent way of managing a connection in SpaceWire: end-to-end flow control and buffer management, fault recovery and packet retransmission are all missing.

QoS: SpaceWire lacks any control over the quality of service provided. For example there is no method for delivering one packet with higher priority than another, or for ensuring deterministic delivery of information.

Fault Detection, Isolation and Recovery: There is no integrated FDIR policy for SpaceWire. While it is simple to cross strap links and provide underlying redundancy thanks to the topological freedom provided by SpaceWire, there is no standard means of managing FDIR.

The SpaceWire-RT protocol must resolve these issues with SpaceWire.

4.2 Coherent Set of Protocols for Spacecraft Applications

SpaceWire-RT has to be able to support all or most spacecraft onboard communication requirements, including:

- Instrument interfacing: connecting an instrument to the mass memory, or to a payload processing unit.
- Device and sub-system networking: interconnecting instruments, mass memory devices, processors, downlink telemetry units, and other on-board electronic equipment.
- Inter-processor communications: supporting multiprocessors data processing systems.
- Gathering housekeeping information: routine, non-critical, collection of equipment status information.
- Deterministic command and control: sending and receiving information within specific time limits, which is important for AOCS/GNC and other control applications.
- Time distribution: distribution of system time to various units for synchronisation or information time-stamping purposes.
- Sub-system synchronisation: synchronisation of two or more units so that they can exchange information or perform data collection or processing together.

- Event signalling: signalling of significant events between units on the network, for example error conditions or service requests.
- Device enumeration: the ability to detect the presence or loss of a unit on the network and have this signalled to an appropriate network manager.

The set of protocols must cover the full range of operational speeds which can be as low a 1 Mbits/s for control applications and up to 20 Gbits/s for data collection from high data-rate instrument like SAR or hyperspectral imagers.

The full range of operational distances must also be covered ranging from chip to chip communication on a board (few cm), to communication between board via a backplane (few tens cm), to inter-unit communication (few m), to communication to specific instruments in remote parts of a spacecraft (few tens m), to long distance communication on launchers or the international space station (over 100 m).

Especially for longer distances, where there can be significant difference in ground potential between units, galvanic isolation is required, to prevent damage to interface devices and significant current flow in the ground connections that can cause noise.

The technology used has to be suitable for space application using radiation tolerant components. The network technology has to be low mass, low power, operate over a wide temperature range, and be rugged enough to withstand the shock and vibration of a launch.

4.3 FDIR

A communication network for use on board a spacecraft has to be robust and durable. It has to have a high reliability and be able to withstand a single point failure. Fault detection, isolation and recovery (FDIR) are essential to provide continued operation in the event of a fault:

- Fault detection is the ability to detect faults occurring in the network
- Fault isolation is the prevention of faults propagating from one network component to another and the prevention of a fault in one data flow affecting another data flow.
- Fault recovery is the recovery of system operation after a fault has occurred. Faults can be temporary, persistent, permanent, or intermittent.
- Fault recovery from a temporary fault simply requires the data unit in which the fault occurred to be resent. If the data unit is large, then a large amount of data will have to be resent and a large amount of buffer storage is required to store that data. This

leads to the data unit being as small as practical, with the limit on small size being any overheads for any necessary data unit head or tail.

- Fault recovery from a persistent error may require the data link or router that the error occurred in to be reinitialised, after which any data lost is resent and normal operation resumed.
- Fault recovery from permanent faults requires redundancy and some form of cross-strapping to be built into the network, so that when there is a failure it is possible first to isolate the faulty unit, and then to activate and connect to a spare unit.
- Fault recovery from an intermittent error requires first for the intermittent error to be correctly diagnosed. Each individual occurrence of an intermittent error will appear like a temporary or persistent error, but if these errors occur frequently enough the effect can be to substantially reduce the available bandwidth of a data link. Once diagnosed, by for example measuring the frequency of errors, the faulty link or unit can be deactivated and a spare unit or data link activated to recovery from the intermittent error as if it were a permanent error. If the intermittent error occurs only infrequently, it may be treated as a temporary or persistent error.

4.4 Quality of Service

Quality of Service (QoS) means that the level of service provided by a communication system can be adjusted to suit specific communication requirements. For example, some data may be essential and if it is lost it must be resent, other data might not be so important and if lost can be replaced next time that a sensor is read routinely, so that no resending of the data is required. The ability to select the service level required for a particular data stream is called Quality of Service.

An important aim of SpaceWire-RT is to be able to provide a full QoS, so that many different types of application can be operated over the same network. In particular the QoS should allow advanced avionics systems and integrated data-handling/avionics systems to be implemented readily.

Features that the SpaceWire-RT QoS has to provide are listed below:

- Responsiveness – ability to react rapidly to real-time events and to deliver information with low latency, which is concerned with network latency, data-rate and priority.

- Determinism – ability to deliver a message and to flag and recover from errors within specific time constraints, which is concerned with network resource reservation.
- Robustness – ability to continue to deliver messages in the event of transitory and permanent faults, which is concerned with acknowledgements, retry mechanisms, redundancy, and autonomous or managed fault detection, isolation and recovery.
- Durability – ability of the network to provide the required network services without intervention for long periods of time, which is concerned with network management, fault isolation and recovery.
- Performance – ability to handle high bandwidth data streams and to provide scalable performance to match application requirements using a range of appropriate communications media enabling power/mass versus performance selection.
- Low latency signalling – highly capable and robust out-of-band signalling techniques integrated within the network to remove the need for additional wires and control/configuration networks. Low latency signalling is required to support event signalling, time distribution and synchronisation.

The QoS for SpaceWire-RT must be integrated within the network, so that each application does not have to develop its own QoS mechanisms which might compete or interfere with those of other applications. A comprehensive set of quality of service capabilities has to be provided by the network, avoiding the need for applications to have to be concerned with network quality. This has the effect of decoupling the applications from the message delivery service, leading to simpler, more reusable, application software.

4.5 Backwards Compatibility with SpaceWire

SpaceWire is widely used on current spacecraft. Many instruments and other units have been designed with SpaceWire interfaces. Software applications have been designed to operate with SpaceWire interfaces and networks. To avoid having to redevelop this equipment and software, the protocols developed for SpaceWire-RT have to be backwards compatible with SpaceWire at some level.

The primary input to a SpaceWire interface or SpaceWire router is a SpaceWire packet, if the SpaceWire-RT protocols adopt the packet level of SpaceWire, much of the existing software can be reused, with the exception of the low-level drivers. The routing concepts of SpaceWire can also be reused so that one of the major advantages of SpaceWire, arbitrary topology networks, is maintained in SpaceWire-RT. The routing concepts include the path and logical addressing mechanisms of SpaceWire.

For lower-level compatibility a bridge between SpaceWire and a particular SpaceWire-RT protocol will be necessary, but this is made much simpler by the adoption of the SpaceWire packet layer by SpaceWire-RT.

5 Evaluation Criteria

In this section the discussion on the key challenges for SpaceWire-RT in section 4, are summarised in a table, forming the evaluation criteria to be used in subsequent trade-offs and analysis. The evaluation criteria are presented in Table 5-1. Note that as would be expected many of these evaluation criteria overlap with the requirements for SpaceWire-RT.

Table 5-1 SpaceWire-RT Protocol Evaluation Criteria

Evaluation Criteria	Description	
Low cable mass	A solution with a low cable mass is preferred. Since a wide range of data-rates and cable lengths need to be supported a solution that supports a range of different cables technologies allowing mass/length selection for a specific application would be preferred.	
High data rate	A solution that supports a range of data rates from 1 Mbit/s to 20 Gbits/s is preferred. If necessary a range of different technologies may be used to achieve the range of data rates.	
Matched impedance	Matched impedance connectors and cables will be essential for the higher data rates.	
Consistent character size	A simple implementation is preferred which might be supported by using consistent character size.	
Parity coverage	To support implementation simplicity parity coverage per character is preferred, if parity is used.	
AC coupled	A solution is preferred which provides AC coupling to support galvanic isolation and fault isolation.	
Robust initialisation	A solution is preferred that has a link initialisation protocol which is robust to ensure that information is not lost when initialisation is performed while a cable is being plugged in, i.e. when connection is temporarily intermittent.	
Galvanic isolation	A solution is preferred which provides galvanic isolation. Not all forms of the link have to provide galvanic isolation.	
Avoid packet blocking	A solution is preferred which avoids packet blocking in	

	routers/links.	
Transport layer	A solution is preferred which provides an end-to-end transport layer protocol.	
Integrated FDIR	A solution is preferred which integrated FDIR in the network, i.e. it is not left up to the end user application to provide network FDIR capability.	
Fault detection	A solution is preferred which can detect the widest range of possible faults.	
Fault isolation	A solution is preferred which isolates as many faults as possible preventing them from propagating from one unit to another or from one data flow to another.	
Fault recovery temporary	A solution is preferred which is able to recover from temporary faults automatically without loss of data.	
Fault recovery persistent	A solution is preferred which is able to recover from persistent faults, that require re-initialisation of the link to recover, automatically without loss of data.	
Fault recovery permanent	A solution is preferred which supports recover from permanent faults. Some loss of data is acceptable.	
Fault recovery intermittent	A solution is preferred which is able to detect and recover from intermittent faults.	
Integrated QoS	A solution is preferred which has quality of service integrated into the network, rather than leaving it up to the application to provide.	
Responsiveness	A solution is preferred which has the ability to react rapidly to real-time events and to deliver information with low latency.	
Determinism	A solution is preferred which has the ability to deliver a message and to flag and recover from errors within specific time constraints.	
Robustness	A solution is preferred which has the ability to continue to deliver messages in the event of transitory and permanent faults.	
Durability	A solution is preferred which has the ability to provide the required network services without intervention for long periods of time.	
Performance	A solution is preferred which is able to handle high bandwidth data streams and to provide scalable performance to match application requirements using a range of appropriate	

	communications media enabling power/mass versus performance selection.	
Low latency signalling	A solution is preferred which incorporates out-of-band signalling techniques within the network to remove the need for additional wires and control/configuration networks.	
Range of applications	A solution is preferred which is able to support the full range of on board communication applications.	
Instrument interfacing	A solution is preferred which can be used to connect instruments to a mass memory or payload processing system.	
Unit networking	A solution is preferred which can be used to interconnect instruments, mass memory, telemetry, processing and other on board data-handling and control units.	
Inter-processor communications	A solution is preferred which can be used to provide multi-processor communication.	
Housekeeping	A solution is preferred which can be used to gather general housekeeping information from instruments, data-handling and control units.	
Event signalling	A solution is preferred which can be used to signal significant events between units on the network.	
Time distribution	A solution is preferred which can be used to distribute system time information to units on the network.	
Synchronisation	A solution is preferred which can be used to synchronise activities between units on the network.	
SpaceWire packets	A solution is preferred which provides a SpaceWire packet interface to the application and which transfers SpaceWire packets across the network from a source node to a destination node.	

6 QoS Mechanisms

In this section the design of a quality of service mechanism for SpaceWire-RT is described, which has been proposed for inclusion in the SpaceFibre standard.

6.1 Frames and Virtual Channels

To provide quality of service, it is necessary to be able to interleave different data flows over a data link or network. If a large packet is being sent with low priority and a higher priority one requests to be sent, it must be possible to suspend sending the low priority one and start sending the higher priority packet. To facilitate this SpaceWire packets are chopped up into smaller data units called frames. When the high priority packet requests to be sent, the current frame of the low priority packet is allowed to complete transmission, and then the frames of the high priority packet are sent. When all the frames of the high priority packet have been sent, the remaining frames of the low priority packet can be sent.

Each frame has to be identified as belonging to a particular data flow so that the stream of packets can be reconstructed at the other end of the link. Low priority packets belong to one data stream and high priority packets belong to another data stream.

Each independent data stream allowed to flow over a data link, is referred to as a virtual channel (VC). Virtual channels are uni-directional and have a QoS attribute, e.g. high priority or low priority. At each end of a virtual channel is a virtual channel buffer (VCB), which buffers the data from and to the application. An output VCB takes data from the application and buffers it prior to sending it across the data link. An input VCB receives data from the data link and buffers it prior to passing it to the receiving application.

There can be several output virtual channels connected to a single data link, which compete for sending information over the link. A medium access controller determines which output virtual channel is allowed to send the next data frame. When an output VCB has a frame of data ready to send and the corresponding input VCB at the other end of the link has room for a full data frame, the output VCB requests the medium access controller to send a frame. The medium access controller arbitrates between all the output VCBs requesting to send a frame. It uses the QoS attribute of each of the requesting VCBs to determine which one will be allowed to send the next data frame.

Priority is one example of a QoS attribute. Other types of QoS are considered in the subsequent sections.

6.2 Precedence

For the medium access controller to be able to compare QoS attributes from different output VCBs, it is essential that they are all using a common measure that can be compared. The name given to this measure is precedence. The competing output VCB with the highest precedence will be allowed to send the next frame.

6.3 Bandwidth Reservation

When connecting an instrument via a network to a mass memory, what the systems engineer needs to know is “how much bandwidth do I have to transfer data from the instrument to the mass memory?” Once the network bandwidth allocated to a particular instrument has been specified, it should not be possible for another instrument to impose on the bandwidth allocated to our instrument. A priority mechanism is not suitable for this application. If an instrument with high priority has data to send it will hog the network until all its data has been sent. In this case what is needed is a mechanism that allows bandwidth to be reserved for a particular instrument.

Bandwidth reservation calculates the bandwidth used by a particular virtual channel, and compares this to the bandwidth reserved for that virtual channel to calculate the precedence for that virtual channel. If the virtual channel has not used much reserved bandwidth recently, it will have a high precedence. When a data frame is sent by this virtual channel, its precedence will drop. Its precedence will increase again over a period of time. If a virtual channel has used more than its reserved bandwidth recently, it will have a low precedence.

A virtual channel specifies a portion of overall Link Bandwidth that it wishes to reserve and expects to use, i.e. its Expected Bandwidth.

When a frame of data is send by any virtual channel, each virtual channel computes the amount of bandwidth that it would have been permitted to send in the time interval that the last frame was sent. This is known as the Bandwidth Allocation. Bandwidth Allowance is calculated as follows:

$$BandwidthAllowance = ExpectedBandwidth \times LastFrameBandwidth$$

Where

> Expected Bandwidth is the portion of overall link bandwidth that a virtual channel wishes to use, and
>
> Last Frame Bandwidth is the amount of data sent in the last data frame.

Each virtual channel can use this to determine its Bandwidth Credit, which is effectively the amount of data it can send an still remain within its Expected Bandwidth. Bandwidth Credit is the Bandwidth Allowance less the Bandwidth Used accumulated over time.

Bandwidth Credit is calculated for each virtual channel as follows:

$$BandwidthCredit = \sum_{Frames} \frac{BandwidthAllowance - UsedBandwidth}{ExpectedBandwidth}$$

Where

> Used Bandwidth is the amount of data sent by a particular virtual channel in the last data frame, which is zero except for all virtual channels except for the one that sent the last frame.

The Bandwidth Credit is updated every time a data frame for any virtual channel has been sent.

A Bandwidth Credit value close to zero indicates nominal use of bandwidth by the virtual channel.

A negative value indicates that the virtual channel is using more than its expected amount of link bandwidth.

A positive value indicates that the virtual channel is using less than its expected amount of link bandwidth.

To simplify the hardware required to calculate the Bandwidth Credit it is allowed to saturate at plus or minus a Bandwidth Credit Limit, i.e. if the Bandwidth Credit reaches a Bandwidth Credit Limit it is set to the value of the Bandwidth Credit Limit.

When the Bandwidth Credit for a virtual channel reaches the negative Bandwidth Credit Limit it indicates that the virtual channel is using more bandwidth than expected. This may be recorded in a status register and used to indicate a possible error condition. A network management application is able to use this information to check correct utilisation of link bandwidth by its various virtual channels.

For a virtual channel supporting bandwidth reserved QoS, the value of the bandwidth counter provides the precedence value for that virtual channel.

The operation of a bandwidth credit counter is illustrated in Figure 6-1.

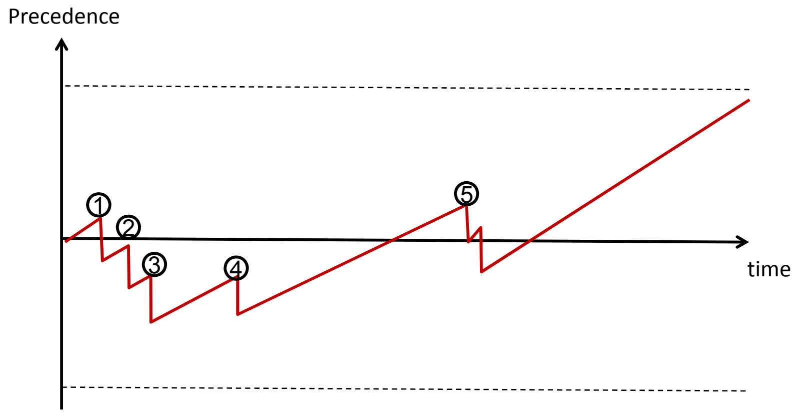

Figure 6-1 Bandwidth Credit Counter

The bandwidth credit for a particular VC increments gradually. At point (1) a frame is sent from by this VC, resulting in a sudden drop in credit. The size of the drop is amount of data sent in the frame divided by the percentage bandwidth reserved for the VC. This means that the smaller the percentage bandwidth the larger the drop, and hence the longer it takes to regain bandwidth credit.

After the drop at point (1) the bandwidth credit gradually increments until point (2) when another frame is sent by the VC. Further frames are sent at points (3), (4), (5) etc. If the frames sent are full frames then the drop in bandwidth credit every time a frame is sent, will be the same size.

The bandwidth credit counter for another VC is illustrated in Figure 6-2. This VC has about half the bandwidth of the VC in Figure 6-1 allocated to it. This means that the drops in bandwidth credit when frames are sent by this VC are about twice the size, as can be seen Figure 6-2 at points (1), (2) and (3).

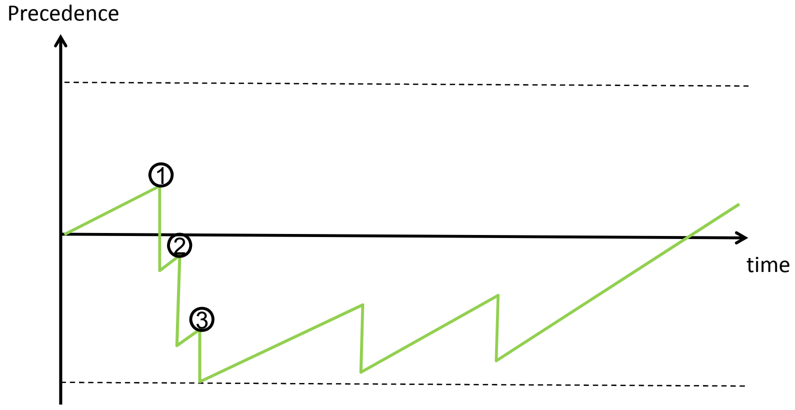

Figure 6-2 Bandwidth Credit Counter with Smaller Reserved Bandwidth

The bandwidth credit counter of another VC is shown in Figure 6-3. In this case the bandwidth credit slowly increments and although some frames are sent at points (1), (2) and (3), the bandwidth credit eventually saturates, reaching it maximum permitted value at point (4). Although more bandwidth should be accumulated after point (4) this is effectively ignored since the maximum possible bandwidth credit has been reached. At point (5) a frame is sent once more, resulting in a drop from the maximum bandwidth credit value.

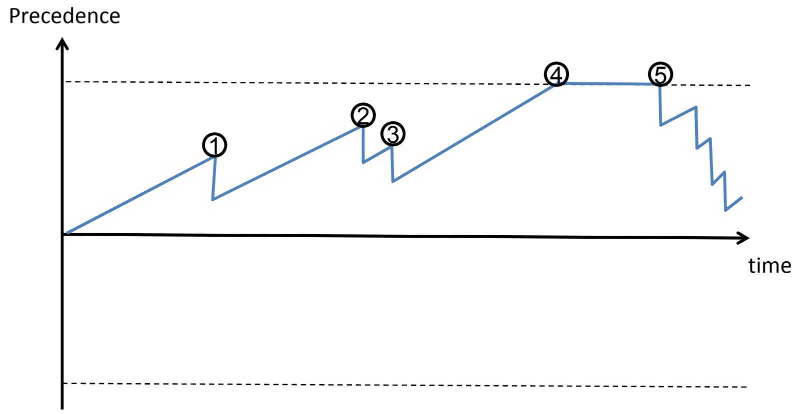

Figure 6-3 Bandwidth Credit Counter Reaching Saturation

All three VCs are shown together in Figure 6-4. When a VC has a data frame ready to send and room for a full data frame at the other end of the link, it competes with any other VCs in a similar state, the one with the highest bandwidth credit being allowed to send the next data frame. At points (1), (2) and (3) the red VC has data to send and sends frames. At points (4), (5) and (6) the green VC has data to send and sends a data frame. At point (7) both the blue and the red VCs have data to send. The blue VC wins since it has the highest bandwidth credit count. After this the red VC is allowed to send a further data frame at point (8).

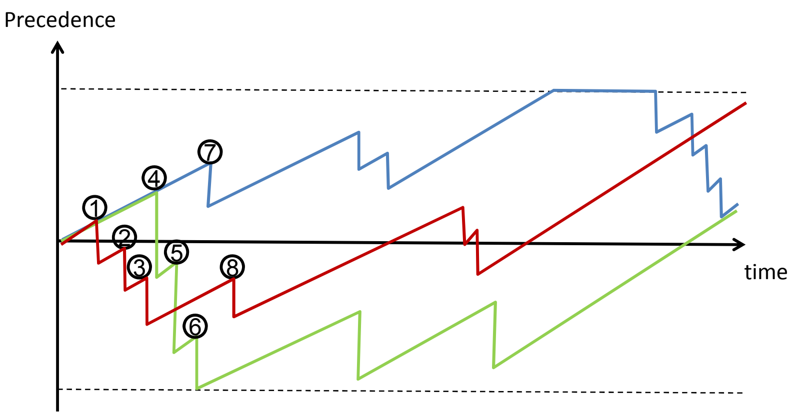

Figure 6-4 Bandwidth Credit Counters of Competing VCs

If the bandwidth credit counter reaches the minimum possible bandwidth credit value, it indicates that it is using more bandwidth than expected. This condition may be flagged to indicate a possible error condition.

Similarly if the bandwidth credit counter stays at the maximum possible bandwidth credit value for a relatively long period of time, the VC is using less bandwidth than expected and this condition can be flagged to indicate a possible error.

The bandwidth credit value is the precedence used by the medium access controller to determine which VC is permitted to send the next data frame.

6.4 Best Effort

The best effort QoS, is permitted to send data when no VC with a different QoS is ready to send data (i.e. has a data frame ready to send and there is room for a full data frame at the far end of the link). This can achieved by simply setting the precedence value below the minimum value permitted for bandwidth credit.

It is possible that two or more best effort VCs are provided and that they become ready to send data at the same time in this case some mechanism is required to be able to select which one is to send the next data frame. Furthermore it is useful in a spacecraft application to be able to detect if a best effort VC is actually sending significantly more or less data than expected.

Since a VC has to be able to support the bandwidth reserved QoS and will therefore contain a bandwidth credit counter, it is possible to use this counter to arbitrate between best effort

VCs that are ready to send, and to provide a mechanism for detecting over or under utilisation of the best effort VC.

For each best effort VC and expected bandwidth figure is provided which indicates how much bandwidth the best effort channel is normally expected to use. The bandwidth credit counter then operates in exactly the same was as for the bandwidth reserved QoS, except that the precedence value is taken as:

$$\mathrm{Pr}\,ecedence = BandwidthCredit - 2 \times MaximumBandwidthCredit$$

This is illustrated in Figure 6-5, which shows the bandwidth reserved VCs from Figure 6-4, together with two best effort VCs, in purple and orange.

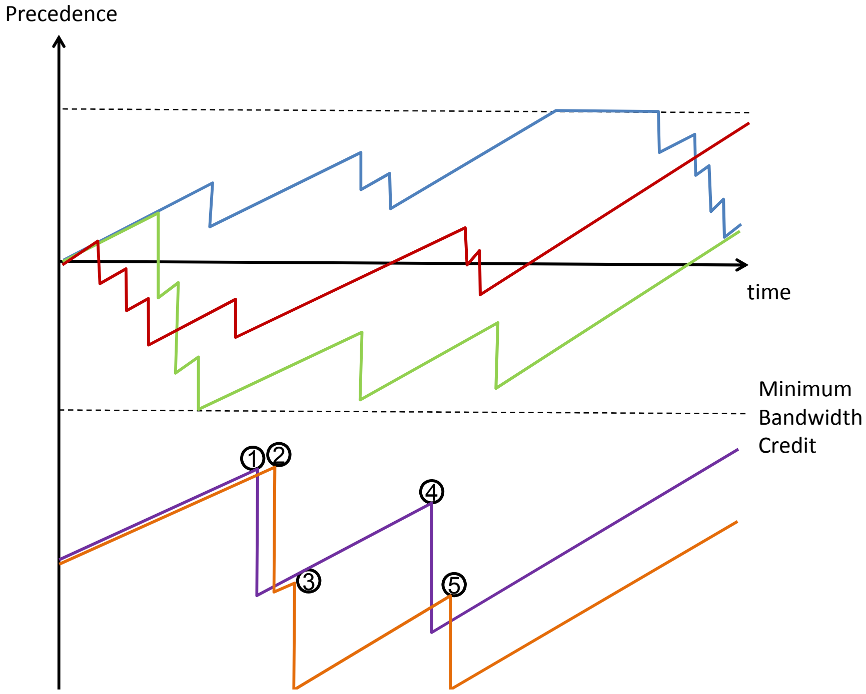

Figure 6-5 Best Effort Quality of Service

All VCs compete for permission to send the next data frame. The one with highest precedence that is ready to send data will win. Since the best effort VCs always have precedence which is less than that of the bandwidth reserved VCs they will only be allowed to send when no bandwidth reserved VCs have data to send.

At point (1) the purple VC is given permission to send a data frame, and its precedence (bandwidth credit) then drops accordingly. At points (2) and (3) the orange VC receives permission to send. At point (4) both the orange and purple VCs are ready to send, but the purple one wins because it has highest precedence. The orange VC subsequently receives permission at point (5).

6.5 Scheduled

To provide fully deterministic data delivery it is necessary for the QoS mechanism to ensure that data from specific virtual channels can be sent (and delivered) at particular times. This can be done by chopping time into a series of time-slots, during which a particular VC is permitted to send data frames.

This is illustrated in Figure 6-6 which shows the bandwidth reserved and best effort precedence values from Figure 6-5, together with the scheduled QoS precedence as a dashed blue line.

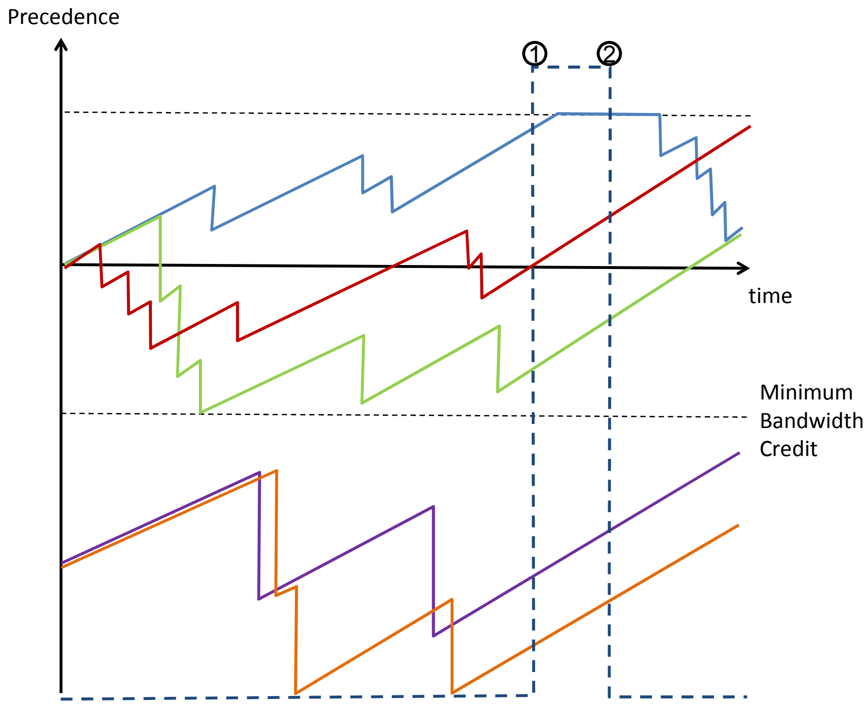

Figure 6-6 Scheduled Quality of Service

The time-slot where the scheduled VC is permitted to send data starts at point (1) and ends at point (2). Outside this interval the scheduled VC is not permitted to send data frames at all. In the time-slot allocated to the VC, the interval between points (1) and (2), the scheduled VC is given higher precedence than any other VC. This means that if the scheduled VC has data ready to send it will always be able to send data frames during its time-slot. If the scheduled VC does not have data to send other VCs will be able to send data according to their precedence value.

Scheduled VCs also contain a bandwidth credit counter which can be used to monitor their use of network bandwidth and to flag possible over or under utilisation of bandwidth by that VC.

Time-slots can be defined by broadcasting start of time-slot signals, or by broadcasting time and having a local time counter which determines the start and end of each time-slot. The SpaceFibre broadcast message mechanism support both synchronisation and time distribution (see section 6.7).

6.6 Priority

The final type of QoS provided by VCs is priority. There are two mechanisms being considered for priority QoS: fixed precedence and multi-layered precedence. One of these will eventually be selected for SpaceFibre and SpaceWire-RT.

6.6.1 Fixed Precedence Priority

Fixed precedence has a set of priority level each of which specify a particular precedence value. The priority levels are arranged as follows:

- Emergency priority: higher precedence than any other QoS.
- Vital priority: higher precedence than any other QoS except emergency priority and scheduled QoS.
- Priority: a range of priorities, each of decreasing precedence, with the high precedence being just less than the maximum bandwidth reserved precedence.

Note: these priority settings are examples and may be refined.

The fixed precedence priority QoS is illustrated in Figure 6-7 where the fixed precedence levels related to each priority level are shown as red dashed lines. These are shown compared to the changing precedence values of the bandwidth reserved, best effort and scheduled QoS from Figure 6-6.

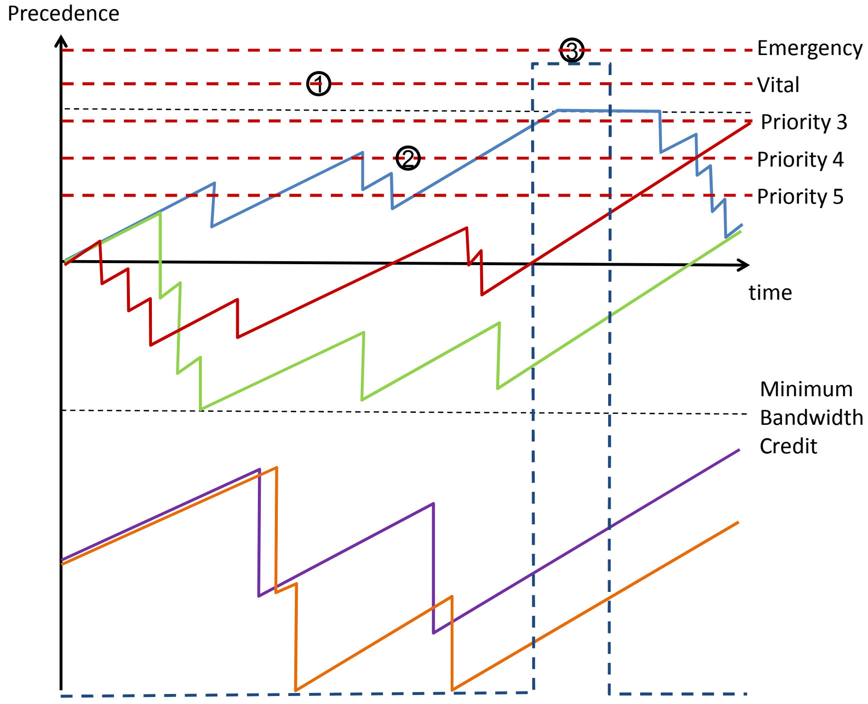

Figure 6-7 Fixed Precedence Priority Quality of Service

Whenever a priority VC is ready to send data and its precedence level is greater than any other VC that has data ready to send, that priority VC is permitted to send the next data frame. At point (1) the vital priority VC has data ready to send and since it always has higher precedence than all the bandwidth reserved and best effort QoS VCs, it is permitted to send the next data frame. At point (2) priority 4 has data ready to send and competes with the bandwidth reserved channels. Since it has higher precedence than the current values of the bandwidth reserved VCs the priority 4 VC can send the next data frame. At point (3) a VC with emergency priority QoS has data ready to send and even though the a VC with scheduled QoS is able to send data, the VC with emergency priority is allowed to send the next data frame.

Each VC set to priority QoS, uses its bandwidth credit counter to monitor the utilisation of the link by that VC comparing it to an expected bandwidth. This can be used to detect priority VCs that are sending much more or much less bandwidth than expected.

6.6.2 Multi-Layered Precedence Priority

The multi-layered precedence priority QoS recognises that with bandwidth reserved and best effort we already have effectively two priority levels: bandwidth reserved VCs will always have priority over best effort VCs. This is then expanded to several priority levels each that operate like bandwidth reserved QoS, as illustrated in Figure 6-8.

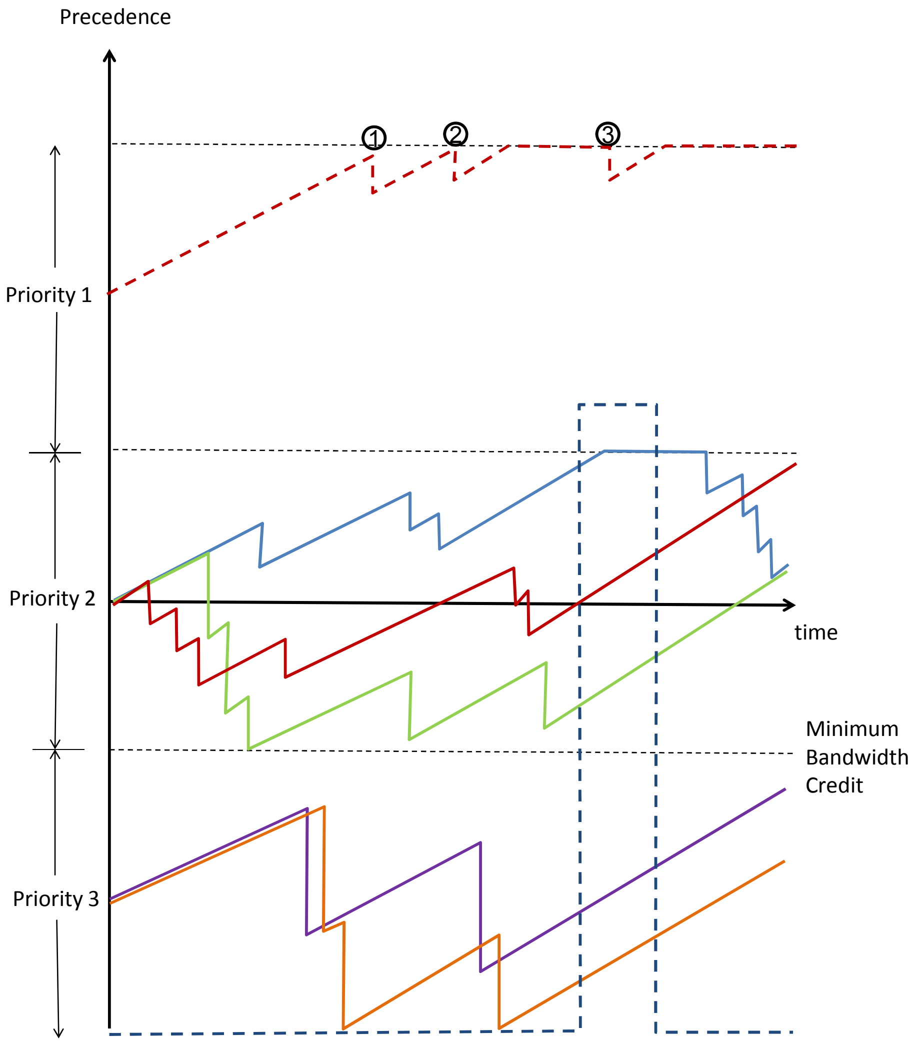

Figure 6-8 Multi-Layered Precedence Priority Quality of Service

In this example there are three priority level, which could be expanded to any reasonable number. Within any level there can be any number of VCs which compete amongst themselves based on their bandwidth credit. A higher priority VC will always have precedence over a lower priority VC.

A single priority 1 VC is shown as a dashed red line. Whenever it has data to send, points (1), (2), and (3), it will be permitted to send the next data frame. The scheduled VC is shown as a priority 2 scheduled VC.

The bandwidth utilisation of every VC can be monitored to detect when it is using much more or much less bandwidth than expected.

The multi-layered precedence priority scheme is preferred for SpaceWire-RT and SpaceFibre as it is much simpler to set the priority levels a priori, to have the required effect on precedence.

6.7 Sideband Signalling Priority

SpaceFibre broadcast messages are short messages containing 8 bytes of data that can be broadcast to every node and router in a network. They are used to provide a high priority "side-band signalling" mechanism, which can be used for time distribution, synchronisation, event signalling and error indication.

Broadcast messages have the highest possible priority in SpaceFibre and can be sent in the middle of a data frame resulting in a low latency signalling mechanism.

7 FDIR Mechanisms

In this section the fault detection, isolation and recovery (FDIR) methods relevant to spacecraft onboard networks are considered.

7.1 Link Level FDIR

A major principle of FDIR is to detect, isolate and recover from a fault as close as possible to the place or time at which the fault occurs. This prevents propagation of the fault across the overall system, but requires suitable FDIR mechanisms to be integrated into the lower levels of the system architecture. For SpaceWire-RT the lowest practical level at which FDIR can be implemented is the link level. This is a natural place to provide FDIR as the links form the interconnections between sub-systems: the boundaries over which faults should not be allowed to propagate.

At the physical level of the link SpaceFibre provides galvanic isolation. This blocks propagation of DC currents which may result from failure at either end of the link.

At the link level transient errors, where a bit or short burst of bits are detected incorrectly, may be caused by electromagnetic interference, or temporary loss of bit synchronisation in the receiver. SpaceFibre aims to contain this type of error in two ways: temporally and spatially.

Temporal containment encapsulates relatively small amounts of data in data frames. A CRC checksum is included in each data frame so that an error in a frame can be detected immediately at the end of that frame. The framing encapsulates small chunks of data isolating the data in one frame from the data in the next. An error in one data frame does not affect data sent in the next data frame. A frame detected as containing an error can then be resent to recover the lost information.

Spatial containment uses independent virtual channels to send different classes of information. A fault on one virtual channel does not affect the flow of data in another virtual channel. The spatial containment in SpaceFibre is conceptually spatial, but is actually implemented by sending frames from different virtual channels at different times over the one physical link.

7.2 System Level FDIR

Typical system level FDIR on board a spacecraft is the responsibility of the overall control computer. A major failure will result in the onboard control computer reporting the fault to the operations centre on Earth and human operators making the decision about remedial action. However, there are some classes of mission where immediate recovery from faults is essential, for example a planetary lander during the descent phase. There is simply not time to send a message to Earth and for corrective information to be returned to the spacecraft.

The need to be able to support autonomous FDIR is essential in some missions. It can also reduce the need for rapid response from ground station staff, allowing more automated operation of ground stations and subsequent reduction in spacecraft operational costs.

The analysis and design of an FDIR system for a spacecraft is outside the scope of the present contract, but is an important area of research for future study.

7.3 Detecting Errors with the QoS Bandwidth Credit Counters

The QoS bandwidth credit counters can be used to detect some important types of error:

Under-utilisation of bandwidth: where a virtual channel has some predefined level of reserved or expected bandwidth and for some reason it does not use all of this expected bandwidth. This may be indicative of an instrument that has ceased to function.

Over-utilisation of bandwidth: where a virtual channel uses more bandwidth than expected. This may be indicative of a “babbling idiot”, an instrument or other sub-system that has failed in a mode where it is continually sending spurious data.

7.3.1 Normal Utilisation of Expected Bandwidth

Under normal operation the bandwidth credit counter hover around the same value: incrementing gradually when no data is being sent and then jumping down when a frame is sent over the virtual channel. If the actual bandwidth used is not exactly the same as the expected bandwidth the bandwidth credit counter will slowly drift in one direction or the other. By deliberately setting the expected bandwidth to be marginally greater than the actual bandwidth the direction of this drift can be made to be upwards. This means that when a stable state is reached the bandwidth credit counter will be at or close to the maximum value, for a given priority level. This is illustrated in Figure 7-1.

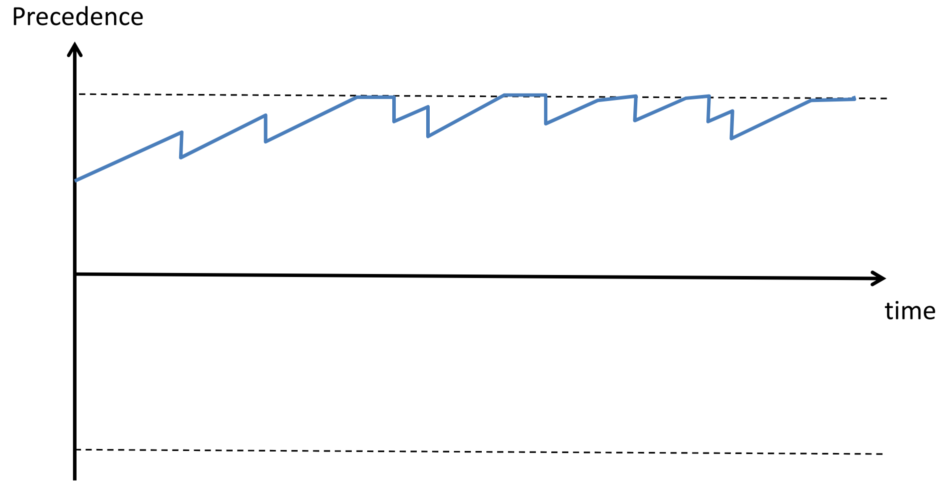

Figure 7-1 Normal Utilisation of Expected Bandwidth

7.3.2 Under Utilisation of Expected Bandwidth

When a virtual channel sends less data than expected the bandwidth credit counter will reach the maximum value for a particular priority level. If it ceases to send data for a significant period of time, the bandwidth credit counter will stay at its maximum value for a long period of time. This can be detected by a timeout timer. The timeout timer would start when the maximum bandwidth credit value is reached. After the timeout timer interval it would expire indicating a possible fault. This is illustrated in Figure 7-2.

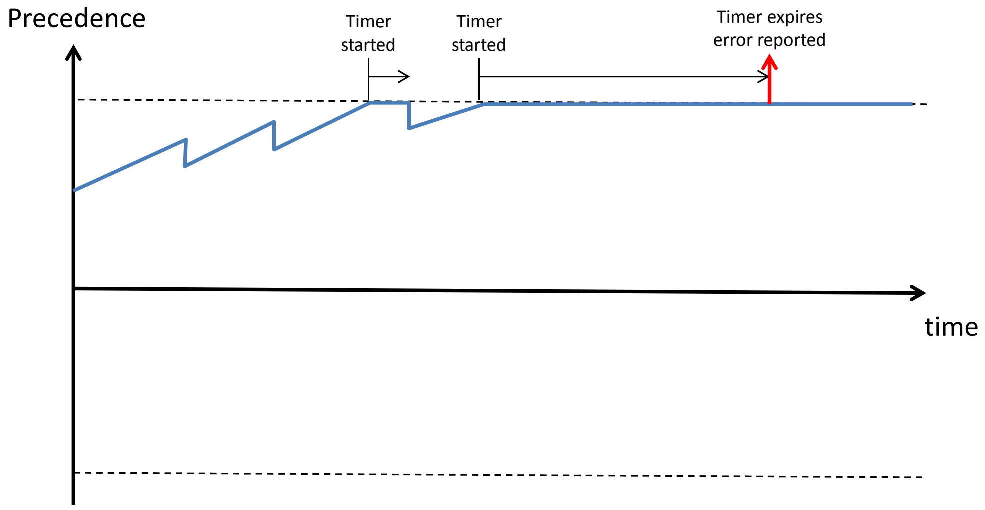

Figure 7-2 Under Utilisation of Expected Bandwidth

This mechanism may be used to detect instruments or other equipment that fails in such a way that it ceases to send data. The indicated error can be signalled to an overall network management controller using a broadcast message. The action taken can then be determined by the network management controller, for example powering down the faulty unit and activating a redundant one.

7.3.3 Over Utilisation of Expected Bandwidth

When a virtual channel uses too much bandwidth, its bandwidth credit counter will rapidly drop. Eventually it will reach the minimum value for a particular priority level. Reaching this minimum value can be used to detect faults when a unit starts to send too much data over a virtual channel: the “babbling idiot” problem. This is illustrated in Figure 7-3. The source is sending data normally and then suddenly starts to send data at every frame where it can. The bandwidth credit counter drops in value as the source blasts out data. By the time the minimum value is reached it is clear that something has gone wrong and a possible fault can be reported to the network management controller. It is the responsibility of SpaceFibre to report the possible fault. It is the responsibility of the network controller to determine the nature of the problem from all the fault reports receive, to evaluate its level of seriousness, and to take appropriate action to recover from the fault. SpaceFibre isolates the fault within the specific virtual channel, or if the link interface itself is faulty, within the link.

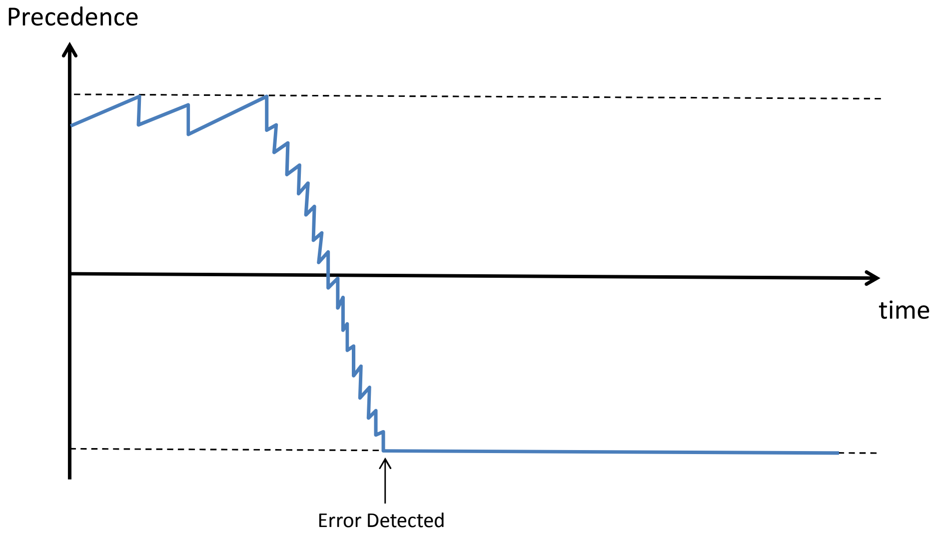

Figure 7-3 Over Utilisation of Expected Bandwidth

8 SpaceFibre

The QoS and FDIR research defined in the SpaceWire-RT project and outlined in sections 6 and 7 have been used in the higher levels of SpaceFibre. In this section SpaceFibre is introduced and then assessed against the requirements detailed and analysed in section 2.1.

8.1 Introduction to SpaceFibre

In this section a brief introduction to SpaceFibre is provided. Full details of SpaceFibre are available in the Draft SpaceFibre Standard "SpaceFibre Draft D, 29th February 2012", published by University of Dundee on the ESA SpaceWire Working Group website http://spacewire.esa.int/WG/SpaceWire/.

SpaceFibre is a very high-speed serial data-link being developed by the University of Dundee for ESA which is intended for use in data-handling networks for high data-rate payloads. SpaceFibre is able to operate over fibre optic and copper cable and support data rates of 2 Gbit/s in the near future and up to 5 Gbit/s long-term. It aims to complement the capabilities of the widely used SpaceWire onboard networking standard: improving the data rate by a factor of 10, reducing the cable mass by a factor of four and providing galvanic isolation. Multi-laning improves the data-rate further to well over 20 Gbits/s.

SpaceFibre will provide a coherent quality of service mechanism able to support best effort, bandwidth reserved, scheduled and priority based qualities of service. It will substantially improve the fault detection, isolation and recovery (FDIR) capability of SpaceWire.

SpaceFibre will support high data-rate payloads, for example synthetic aperture radar and hyper-spectral optical instruments. It will provide robust, long distance communications for launcher applications and will support avionics applications with deterministic delivery constraints through the use of virtual channels. SpaceFibre will enable a common onboard infrastructure to be used across many different mission applications resulting in cost reduction and design reusability. SpaceFibre can run over fibre optic or copper cables.

An overview of the SpaceFibre CODEC architecture is provided in Figure 8-1.

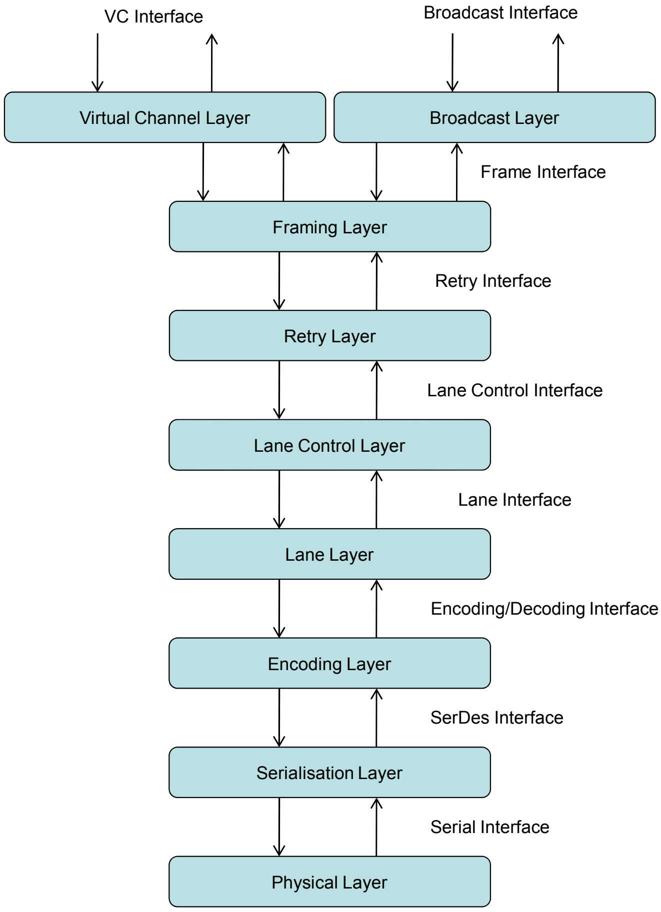

Figure 8-1 Overview of SpaceFibre CODEC

There are nine conceptual layers to the SpaceFibre CODEC:

- Virtual Channel and Flow Control: responsible for quality of service and flow control over the SpaceFibre link.
- Broadcast: responsible for broadcasting short messages across a SpaceFibre network and for receiving and checking those messages.
- Framing: responsible for framing SpaceWire packet data, broadcast messages and FCTs to be sent over the SpaceFibre link. It is also responsible for scrambling SpaceWire packet data for EMC mitigation purposes.
- Retry: responsible for recovering from transient and persistent errors on the SpaceFibre link, and for reporting errors and link failure. Detects missing and out of sequence frames.

- Lane Control: responsible for operating several SpaceFibre lanes in parallel to provide a higher data throughput and to provide redundancy with graceful degradation.
- Lane: responsible for initialising the lane, detecting lane errors and re-initialising the lane after an error has been detected.
- Encoding/Decoding: responsible for encoding data into symbols for transmission and decoding symbols into data for reception.
- Serialisation: responsible for serialising and de-serialising SpaceFibre symbols so that they may be transferred over the physical medium.
- Physical: responsible for transferring the electrical signals across a fibre optic or copper medium.

8.2 Reviewing SpaceFibre against the SpaceWire-RT Requirements

In this section SpaceFibre is reviewed against the requirements for SpaceWire-RT that have been summarised in section 2.1. Requirements that are not met by SpaceFibre are shaded red. Requirements that are not specified directly by SpaceFibre but could be implemented in SpaceFibre Routers are shaded yellow. All other requirements are met by SpaceFibre and are not shaded.

Table 8-1: SpaceFibre Against SpaceWire-RT Requirements

Req#	Title	Requirement	SpaceFibre
10	Data Rate (data handling)	Shall be capable of data rates up to 20 Gbits/s.	Yes, SpaceFibre supports data rates of 2 Gbits/s per lane and will support 5 Gbits/s in future. Multi-laning of upto 10 lanes provides the required maximum data rate.
11	Data Rate (all others)	Shall be capable of data rates up to 400 Mbits/s.	Yes, SpaceFibre supports data rates of 2 Gbits/s per lane.
20	Distance (control bus)	Shall operate over a distance of up to 100 m.	Yes, SpaceFibre supports fibre optic communication over

			distances of 100m.
21	Distance (all others)	Shall operate over a distance of 1 m to 10 m.	Yes, SpaceFibre supports communication over copper cables of distances up to 5 m and fibre optic communication for longer distances.
30	Galvanic isolation (control bus)	Shall provide galvanic isolation for long distant applications.	Yes, SpaceFibre provides capacitive coupling for electrical signals. Transformer coupling might also be developed. Optical communication provides full galvanic isolation.
31	Galvanic isolation (all others)	Should provide galvanic isolation for in-box applications.	Yes, SpaceFibre provides capacitive coupling.
40	Transmission medium (data handling, control bus, computer bus)	Shall operate on twisted pair, co-ax or optical fibre.	Yes, SpaceFibre uses current mode logic (CML) for communication over 100 ohm differential impedance copper media, which include PCB tracks, twisted pairs and coax pairs. SpaceFibre also runs over optical fibre.
41	Transmission medium (telemetry bus)	Shall operate on twisted pair.	Yes, see above.
50	packet size (data handling, computer bus)	Shall support application packet sizes up to at least 32 Mbytes.	Yes, SpaceFibre supports SpaceWire packet transfer with arbitrary sized packets.
51	packet size (control bus, telemetry)	Shall support packet sizes in the range from 8 bytes to 64 Kbytes.	Yes, SpaceFibre supports SpaceWire packet transfer with arbitrary sized packets.
60	Maximum latency (control bus)	Shall support a maximum latency of less than 100 µs.	Yes, for control applications SpaceFibre provides deterministic data delivery using a scheduling mechanism.
61	Maximum latency (time synchronization bus)	Shall support a maximum latency of up to 100 ns.	No, it is not possible for SpaceFibre to provide this very demanding latency requirement. A latency of around 1 µs per link in a network, including router

			broadcast, is possible.
62	Maximum latency (computer bus)	Should support a maximum latency of less than 100 ns over a single link.	No, for data transfer the data rate is likely to be of the order of 1 µs per link in a network, including router broadcast.
70	Reliability	Shall provide a capability for reliable data delivery.	Yes, SpaceFibre provides a retry mechanism for recovery from transient and persistent errors.
80	Determinism	Shall provide determinism.	Yes, SpaceFibre provides deterministic data delivery using a scheduling mechanism.
90	Validity	Shall support a message bit error rate of less than 10-15.	Yes, SpaceFibre basic error rate is assumed to be 10-12. It is likely that this error rate will be significantly lower than this (TBC). Error detection using disparity, invalid codes and CRC ensures that errors are detected and a retry mechanism recovers from the errors automatically, giving an error rate acceptable for space missions, which is well below 10-15.
100	Automatic acknowledgement (control bus)	Should support configurable automatic acknowledgement.	No, SpaceFibre does not provide end to end acknowledgement this would require an additional protocol level. For example, SpaceWire RMAP could be used over SpaceFibre to provide this capability.
110	Automatic fault detection	Shall support automatic fault detection.	Yes, SpaceFibre includes disparity, invalid symbol and CRC protection.
111	Automatic fault identification	Should support automatic fault identification.	Yes, SpaceFibre includes the recording of fault information to support fault identification.
120	Failure and fault tolerance of network (data handling network)	Network should be able to automatically recover from faults.	Yes, SpaceFibre recovers automatically from transient and persistent faults.

			No, SpaceFibre does not support automatic recovery of a network from a permanent link failure. It does support notification of a permanent failure to a network management system.
130	Multi-path transmission (control bus)	Shall support multi-path transmission.	SpaceFibre supports multi-path transmission over a network. However, it is up to the sending node to provide the two copies of a packet.
140	Broadcast data transfer (time synchronisation bus)	Shall support broadcast data transfer.	Yes, SpaceFibre provides a specific time distribution broadcast mechanism to support time information.
150	Multi-cast data transfer (computer bus)	Shall support multi-cast data transfer.	No, SpaceFibre does not support multi-cast. This is a routing function and could be included in a SpaceFibre router.
160	Out-of-band signals	Shall transfer synchronization signals and interrupts with very short latency.	Yes, SpaceFibre provides broadcast messages for distributing “out-of-band” signals.
170	Mass interconnect	Shall be less than 30 g/m (for one lane).	Yes, the copper cable mass for SpaceFibre is yet to be determined but is likely to be in this region. SpaceFibre optical cable is also expected to be around 30 g/m.
180	Power consumption	Should be less than 200 mW	SpaceFibre uses CML which is expected to provide a power consumption in this region, TBC. Current devices are significantly higher power consumption than 200 mW (e.g. TLK2711 SerDes).
190	Communication	Shall support the communication requirements as described in Table 3-2.	See individual requirements below.
200	Data-Handling Distance	Shall support communication over distances of a few cm to 100m.	Yes, SpaceFibre supports the full range of distances required using PCB traces, copper cable (up to

			5 m) and optical fibre (up to 100m).
201	Data-Handling Rate	Shall provide communication data rates of 1 Mbits/ to 20 Gbits/s.	Yes, SpaceFibre can support up to 20 Gbits/s using multiple lanes. At present a single lane will run at 2 Gbit/s which is expected to increase to 5 Gbits/s in future.
202	Data-Handling Latency	Shall support a maximum end to end latency of less than 100 µs over a distance of 100 m and through 10 routing switches.	Yes, SpaceFibre will provide this level of latency over a large network.
203	Data-Handling Packet Size	Shall support packet sizes ranging from 8 bytes to 32 Mbytes.	Yes, SpaceFibre can support SpaceWire packets of any size.
204	Data-Handling QoS	Shall provide reserved bandwidth QoS.	Yes, SpaceWire provides a bandwidth reservation mechanism in hardware. This can be applied to a virtual point-to-point link from instrument to mass memory (for example).
210	Control Bus Distance	Shall support communication over distances of a few cm to 100m.	Yes.
211	Control Bus Rate	Shall provide communication data rates of up to 10 Mbits/s.	Yes.
212	Control Bus Latency	Shall support a maximum end to end latency of less than 100 µs over a distance of 100 m and through 10 routing switches.	Yes.
213	Control Bus Packet Size	Shall support packet sizes ranging from 8 bytes to 64 kbytes.	Yes.
214	Control Bus QoS	Shall provide deterministic QoS.	Yes, SpaceFibre provides a schedule communication mechanism for supporting deterministic data delivery.
220	Telemetry (HK) Distance	Shall support communication over distances of a few cm to 100m.	Yes, for individual link distances over 5m fibre optic cable is required.
221	Telemetry (HK) Rate	Shall provide communication data rates	Yes.

		of up to 10 Mbits/s.	
222	Telemetry (HK) Latency	Shall support a maximum end to end latency of less than 100 μs over a distance of 100 m and through 10 routing switches.	Yes.
223	Telemetry (HK) Packet Size	Shall support packet sizes ranging from 8 bytes to 1 Mbytes.	Yes.
224	Telemetry (HK) QoS	Shall provide reserved bandwidth QoS.	Yes, SpaceFibre provides a reserved bandwidth QoS which can be applied to a virtual network.
230	Computer Bus Distance	Shall support communication over distances of a few cm to 1m.	Yes, SpaceFibre can support data transfer over PCB backplanes.
231	Computer Bus Rate	Shall provide communication data rates of up to 20 Gbits/s.	Yes, using multiple lanes.
232	Computer Bus Latency	Shall support a maximum link latency of less than 100 ns over a distance of 1 m.	No, SpaceFibre cannot achieve this low latency. It is expected that latency over a link will be less that 1 μs.
233	Computer Bus Packet Size	Shall support packet sizes ranging from 8 bytes to 32 Mbytes.	Yes.
234	Computer Bus QoS	Shall provide reserved bandwidth QoS.	Yes, SpaceFibre can support reserved bandwidth for virtual networks. Multiple virtual networks can run in parallel connecting different computers to other computers, memory or IO.
240	Time-Sync Bus Distance	Shall support communication over distances of a few cm to 100m.	Yes.
241	Time-Sync Bus Rate		Yes, SpaceFibre provides a dedicated time-distribution broadcast mechanism.
242	Time-Sync Bus Latency	Shall support a maximum link latency of less than 100 ns over a distance of 10 m.	No, SpaceFibre cannot achieve this low latency. It is expected that latency over a link will be less that 1 μs.

243	Time-Sync Bus Jitter	Shall have link latency jitter of less than 100 ns.	SpaceFibre is likely to be able to achieve this level of performance over a router (TBC).
244	Time-Sync Bus Data Size	Shall support transfer of up to 8 bytes of time information.	Yes, SpaceFibre distributes time information using a broadcast message containing CCSDS unsegmented time.
245	Time-Sync Bus QoS	Shall provide high priority QoS.	Yes, SpaceFibre broadcast messages have highest priority.
250	Sideband Bus Distance	Shall support communication over distances of a few cm to 100m.	Yes.
251	Sideband Bus Rate		Yes, SpaceFibre provides a broadcast mechanism for sending sideband signals.
252	Sideband Bus Latency	Shall support a maximum end to end latency of less than 10 μs over 100 m and across a network comprising 10 routers i.e. less than 1 μs per link including distribution by a router.	No, SpaceFibre cannot achieve this low latency. It is expected that latency over a link will be less that 1 μs.
253	Sideband Bus Packet Size	Shall support transfer of up to 8 bytes of error or interrupt information.	Yes, broadcast messages contain 8 bytes of data.
254	Sideband Bus QoS	Shall provide high priority QoS.	Yes, SpaceFibre broadcast messages have highest priority.

From the analysis reported in the table above it is clear that SpaceFibre would meet all of the SpaceWire-RT requirements with the exception of latency for computer bus, time-synchronisation and sideband signalling. SpaceFibre is able to provide a latency of around 1 μs per link which is likely to be adequate for most spacecraft applications.

Other minor issues are:

- Multipath
- Multicast
- Automatic Acknowledgement
- Automatic Recovery of Network from Permanent Link (or Router) Failure

9 Evaluating SpaceFibre

SpaceFibre meets most of the requirements for SpaceWire-RT but when compared against the evaluation criteria of section 5 fails to meet the full range of capabilities outlined in the evaluation criteria.

Table 9-1 SpaceFibre vs Evaluation Criteria

Evaluation Criteria	Description	SpaceFibre
Low cable mass	A solution with a low cable mass is preferred. Since a wide range of data-rates and cable lengths need to be supported a solution that supports a range of different cables technologies allowing mass/length selection for a specific application would be preferred.	Yes
High data rate	A solution that supports a range of data rates from 1 Mbit/s to 20 Gbits/s is preferred. If necessary a range of different technologies may be used to achieve the range of data rates.	Does not cover lower rates efficiently
Matched impedance	Matched impedance connectors and cables will be essential for the higher data rates.	Yes
Consistent character size	A simple implementation is preferred which might be supported by using consistent character size.	Yes
Parity coverage	To support implementation simplicity parity coverage per character is preferred, if parity is used.	Yes
AC coupled	A solution is preferred which provides AC coupling to support galvanic isolation and fault isolation.	Yes
Robust initialisation	A solution is preferred that has a link initialisation protocol which is robust to ensure that information is not lost when initialisation is performed while a cable is being plugged in, i.e. when connection is temporarily intermittent.	Yes
Galvanic isolation	A solution is preferred which provides galvanic isolation. Not all forms of the link have to provide galvanic isolation.	Yes
Avoid packet blocking	A solution is preferred which avoids packet blocking in routers/links.	Yes, using virtual channels
Transport layer	A solution is preferred which provides an end-to-end transport layer protocol.	Not included in current SpaceFibre

		specification.
Integrated FDIR	A solution is preferred which integrated FDIR in the network, i.e. it is not left up to the end user application to provide network FDIR capability.	Yes
Fault detection	A solution is preferred which can detect the widest range of possible faults.	Yes
Fault isolation	A solution is preferred which isolates as many faults as possible preventing them from propagating from one unit to another or from one data flow to another.	Yes
Fault recovery temporary	A solution is preferred which is able to recover from temporary faults automatically without loss of data.	Yes
Fault recovery persistent	A solution is preferred which is able to recover from persistent faults, that require re-initialisation of the link to recover, automatically without loss of data.	Yes
Fault recovery permanent	A solution is preferred which supports recover from permanent faults. Some loss of data is acceptable.	No
Fault recovery intermittent	A solution is preferred which is able to detect and recover from intermittent faults.	No
Integrated QoS	A solution is preferred which has quality of service integrated into the network, rather than leaving it up to the application to provide.	Yes
Responsiveness	A solution is preferred which has the ability to react rapidly to real-time events and to deliver information with low latency.	Yes
Determinism	A solution is preferred which has the ability to deliver a message and to flag and recover from errors within specific time constraints.	Yes
Robustness	A solution is preferred which has the ability to continue to deliver messages in the event of transitory and permanent faults.	SpaceFibre does not cover recovery from permanent failure.
Durability	A solution is preferred which has the ability to provide the required network services without intervention for long periods of time.	Yes, except for recovery from permanent failure.
Performance	A solution is preferred which is able to handle high bandwidth data streams and to provide scalable performance to match application requirements using a range of appropriate communications media enabling power/mass versus performance selection.	SpaceFibre does not cover slower data rates efficiently

Low latency signalling	A solution is preferred which incorporates out-of-band signalling techniques within the network to remove the need for additional wires and control/configuration networks.	Yes
Range of applications	A solution is preferred which is able to support the full range of on board communication applications.	SpaceFibre does not cover lower data rate applications efficiently.
Instrument interfacing	A solution is preferred which can be used to connect instruments to a mass memory or payload processing system.	Yes
Unit networking	A solution is preferred which can be used to interconnect instruments, mass memory, telemetry, processing and other on board data-handling and control units.	Yes
Inter-processor communications	A solution is preferred which can be used to provide multi-processor communication.	Yes
Housekeeping	A solution is preferred which can be used to gather general housekeeping information from instruments, data-handling and control units.	Yes
Event signalling	A solution is preferred which can be used to signal significant events between units on the network.	Yes
Time distribution	A solution is preferred which can be used to distribute system time information to units on the network.	Yes
Synchronisation	A solution is preferred which can be used to synchronise activities between units on the network.	Yes
SpaceWire packets	A solution is preferred which provides a SpaceWire packet interface to the application and which transfers SpaceWire packets across the network from a source node to a destination node.	Yes

From the above evaluation of SpaceFibre against the evaluation criteria, it is apparent that SpaceFibre scores very highly against the evaluation criteria with main three points that it fails to meet:

1. Efficient implementation of lower data rates: SpaceFibre is aimed at high-speed communication on board spacecraft. To meet the needs of SpaceWire-RT SpaceFibre needs to be extended to cover lower data rate operation efficiently and in a range of implementation technologies. SpaceWire covers these lower data rates

but has many required attributes of SpaceWire-RT missing. Some sort of cross between SpaceFibre and SpaceWire might be an attractive solution for SpaceWire-RT at lower data rates.

2. End to end transport layer protocol. SpaceFibre, as currently, defined has concentrated on the link level protocol, using SpaceWire networking and packet layer protocols. SpaceWire Remote Memory Access Protocol (RMAP) is one higher layer protocol that provides transport layer functionality. RMAP can run over SpaceFibre. It may be appropriate to devise a complete range of end to end transport layer protocols for SpaceWire-RT.

3. Automatic recovery from intermittent and permanent faults. SpaceFibre is able to recover from transient and persistent faults, but does not currently provide a mechanism for recovery neither from permanent faults, nor for detection and recovery from intermittent faults. Intermittent faults are effectively frequently occurring temporary or persistent faults that significantly impact on the operation or bandwidth available from a communication link. One of the main reasons that automatic recovery from intermittent and permanent faults was not included in SpaceFibre was because the user community had a strong aversion to automatic fault recovery, preferring for this type of decision to be left to a remote human operator. For some Earth orbiting missions this may be appropriate, but for many interplanetary missions or for mission critical services an automatic mechanism for fault recovery is preferred. The question then arises whether this should be part of the network responsibility or part of some other spacecraft function. One of the principles of successful fault management is to localise and contain the fault as soon as possible. This can best be done within the network itself. A set of automatic fault recovery mechanisms should be provided for SpaceWire-RT.

10 Potential Solutions

When comparing SpaceFibre to the SpaceWire-RT requirements and evaluation criteria it is clear that SpaceFibre is a very attractive solution lacking just a few features and needing to be extended to lower frequency operation. The missing features are listed below in order of importance for SpaceWire-RT.

1. Efficient implementation of lower data rates: An efficient and versatile means of providing lower data rates is essential to the aims of SpaceFibre and should be the main design activity of the SpaceWire-RT project.
2. Latency for computer bus, time-synchronisation and sideband signalling: Latency comprises several parts, latency on the line, latency in the CODEC, and latency in the routers forming the network. Latency on the line is limited by the laws of physics. Latency in the CODEC is concerned with where the sideband signals are injected into the data stream, and the degree of pipelining in a specific implementation. Network latency is concerned with the time it takes for a sideband signal to be received, validated and forwarded out of a router, and the number of routers traversed across a network. Work in the SpaceWire-RT project will focus on reviewing the low level design of the SpaceFibre CODEC specification and implementation with the aim of identifying any opportunity for reducing latency.
3. Automatic recovery from intermittent and permanent faults. This is an important feature of a fault tolerant network, although it may not be accepted by some spacecraft engineers. Methods for implementing automatic recovery from intermittent and permanent faults will be considered in the SpaceWire-RT project if time permits.
4. End to end transport layer protocol. SpaceFibre defines communication across a link, with SpaceWire routing being used to forward a packet over a network. Flow control is across a link. Depending on the router implementation a virtual channel can provide end to end flow-control if the virtual channels of the individual links are concatenated to form a virtual channel. This approach will be considered further in the SpaceWire-RT project if time permits.
5. Multipath: the sending of the same message over two or more different paths through a network is a means of providing communication redundancy. This is simple to implement in principle, but is only applicable to some missions because of the doubling of communication cost. The SpaceFibre broadcast message does provide

multipath communication, broadcasting a message over all active links in a network. Multipath will only be considered further in the current project if time permits.

6. Multicast: The forwarding of messages to more than one destination can be achieved by simply sending the message several times each to one of the required different destinations. Multicast can also be implemented in a router. Multicast will only be considered further in the current project if time permits.
7. Automatic acknowledgement: while a useful feature for SpaceWire-RT it is only required by certain classes of application. When required this type of facility can be implemented in the application without too much difficulty. It will only be considered further in the current project if time permits.

From the above it is clear that the main focus of the next stage of the SpaceWire-RT project should be concerned with extending SpaceFibre to operate efficiently at lower data rates. To this end a range of possible SpaceFibre related protocols that provide different communication characteristics are presented for consideration. Each uses the upper layers of SpaceFibre, which provide QoS, FDIR and laning, replacing all or part of the lower layers (lane, encoding, serialisation, physical).

1. SpaceFibre-CML
2. SpaceFibre-LVDS
3. SpaceFibre-Oversampled-LVDS
4. SpaceFibre over SpaceWire
5. SpaceFibre over SpaceWire 8B/12B DS

SpaceFibre-CML is standard SpaceFibre, which is called SpaceFibre-CML here to differentiate it from SpaceFibre-LVDS. SpaceFibre-CML is SpaceFibre running over Current Mode Logic (CML) which can operate over copper or fibre optic cable.

10.1 SpaceFibre-LVDS

SpaceFibre-LVDS is SpaceFibre which uses Low Voltage Differential Signalling rather than CML. LVDS is a particular form of CML which has been used in spacecraft applications. It is restricted to data rates up to several hundred Mbits/s, likely to be around 600 Mbits/s for space qualified components.

SpaceFibre-LVDS requires phase-locked loop (PLL) or similar technology to be implemented in the CODEC for clock recovery. The slower operating speed makes Delay Locked Loop technology feasible which is simpler to implement is digital technology.

Advantages:

- LVDS interfaces available on most FPGAs
- LVDS proven in space flight
- Minor modification to SpaceFibre standard
- May save some power compared to CML (TBC)
- Lower cable mass than SpaceWire
- Covers 10 Mbits/s to 600 Mbits/s speed range

Disadvantages:

- Requires PLL or similar clock recovery circuitry
- Not possible to implement in current space qualified FPGAs

10.2 SpaceFibre-Oversampled-LVDS

For lower speed operation, below around 100 Mbits/s, it is possible to use an oversampling technique to recover the receive clock from the SpaceFibre bit stream. When coupled with LVDS technology this offers a SpaceFibre solution that can be implemented in current radiation tolerant FPGA technology. It has the advantage of requiring fewer wires than SpaceWire and is galvanically isolated.

Figure 10-1 illustrates the oversample technique for data recovery from a serial bit stream. This technique assumes that the data is being transmitted at a known bit rate and that a clock at the same frequency is available in the receiver. Furthermore the transmit bit rate must not drift significantly from its nominal bit rate.

In Figure 10-1 (a), the eye pattern of a typical serial bit stream is illustrated. An eye mask indicates the minimum amplitude that the receiver recognises (A). It also indicates the acceptable region for sampling the received signal (W). Sampling of the received signal should ideally take place in the middle of the eye, but because of jitter in the sampling clock there is some uncertainty in when it will be sampled.

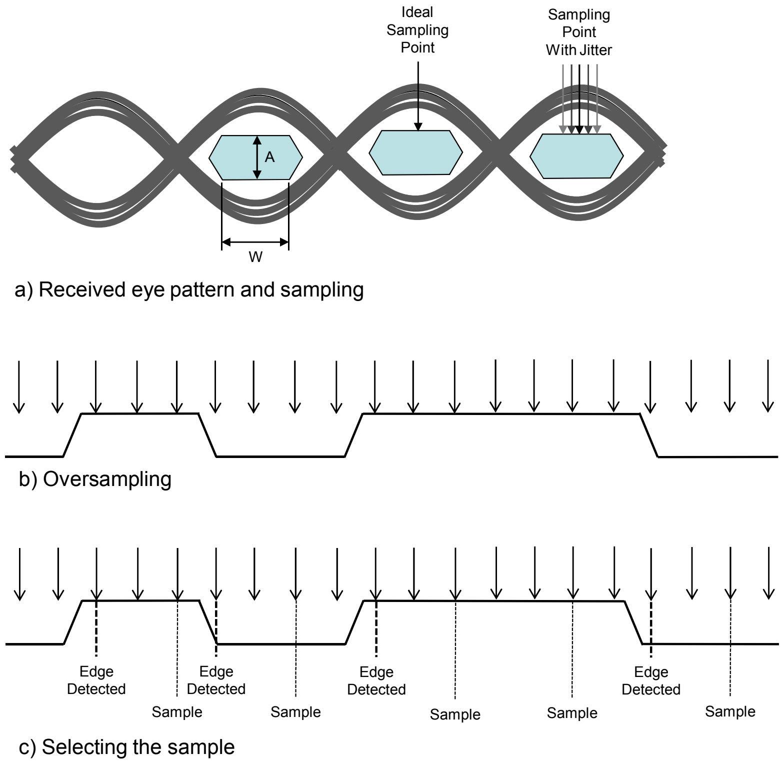

Figure 10-1 Oversampling Clock and Data Recovery

The oversampling technique samples the received data signal at a higher frequency than the bit rate, for example four times the bit rate. This is illustrated in Figure 10-1 (b) where an example bit stream carrying data 010110 is shown. There are nominally four samples per bit interval, although due to signal and clock jitter this may vary.

To determine which sample to use as the recovered data, the edges in the data stream are first detected. This is straightforward: a 0 sample followed by a 1 sample indicates that a rising edge has occurred, and a 1 sample followed by a 0 sample indicates that a falling edge occurred. Since there are nominally four samples in a bit interval the data should be recovered using the sample point two samples after an edge was detected. If there is no edges the data is recovered every four samples from the last edge. If an 8B/10B encoding scheme is being followed which ensures several edges in each group of 10 bits the oversampling scheme is able to track drift in the transmit clock, provided that it remains close to the nominal bit rate.

Oversampling is an entirely digital data recovery technique and does not need any special analogue circuitry. It requires the digital input signal to be sampled at four times the bit rate or higher. The sampling can be done with a clock running at four times the bit rate or at the same speed as the bit rate using a four phase clock and sampling on each of the phases.

Details of one possible oversampling technique are given in a Xilinx application note; Nick Sawyer, “Data Recovery”, XAPP224 (v2.5), Xilinx, July 11, 2005.

Advantages:

- LVDS interfaces available on most FPGAs
- LVDS proven in space flight
- Minor modification to SpaceFibre standard
- May save some power compared to CML (TBC)
- Lower cable mass than SpaceWire
- Clock recovery does not require PLL
- Covers 1 Mbits/s to 100 Mbits/s (TBC) speed range
- Can interoperate with SpaceFibre-LVDS depending on speed used

Disadvantages:

- Operation at over 100 Mbits/s may be difficult depending on the specific implementation technology. Xilinx Virtex II FPGAs support data recovery at 400 Mbits/s with oversampling. More recent Xilinx FPGAs are expected to run faster.

10.3 SpaceFibre over SpaceWire

The lane layer of SpaceFibre could be replaced by SpaceWire, with SpaceFibre frames being embedded in SpaceWire packets.

Note: this approach was suggested by Professor Masaharu Nomachi of Osaka University and also follows from the earlier work on SpaceWire-RT by University of Dundee for ESA.

SpaceFibre over SpaceWire has the advantage that it uses existing flight qualified SpaceWire technology, enhancing it with the QoS, FDIR and laning capabilities of SpaceFibre. The corresponding protocol stack is illustrated in Figure 10-2.

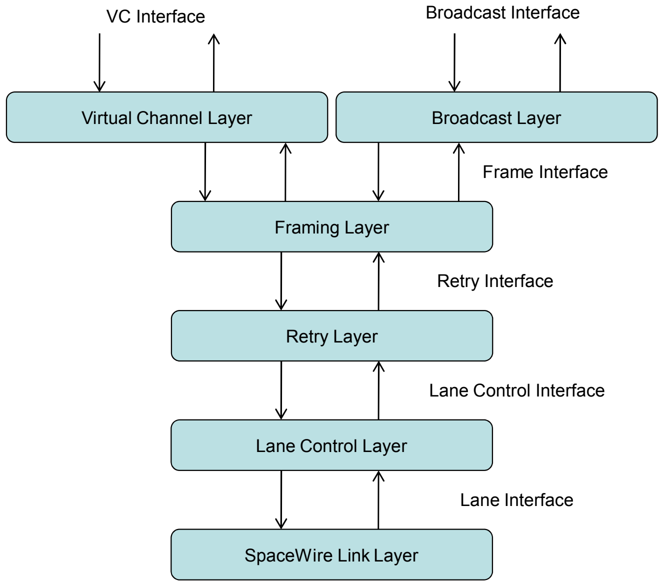

Figure 10-2 SpaceFibre Over SpaceWire

The SpaceFibre lane, encoding, serialisation and physical layers are replaced by the SpaceWire Link Layer, which is standard SpaceWire. The user application passes SpaceWire packets to be sent over the network into the virtual channel layer. These packets are segmented into frames by the framing layer. The frames are passed to the SpaceWire link level where they are encapsulated into individual SpaceWire packets. Since the maximum size of a SpaceWire packet is 66 words (264 bytes) the SpaceWire packets are all small. Multiple application packets can be multiplexed over a single SpaceWire packet using the SpaceFibre virtual channels and medium access controller.

Figure 10-3 shows a SpaceWire packet embedded in SpaceWire packets.

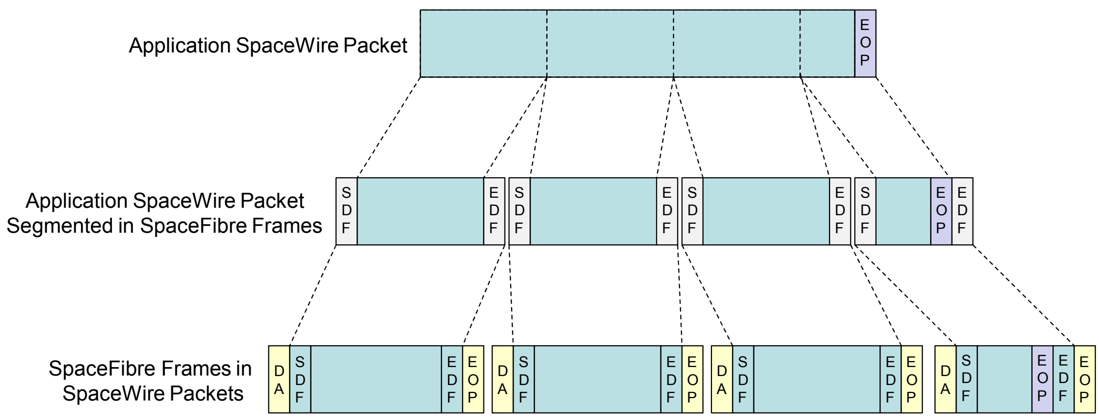

Figure 10-3 Application SpaceWire Packet in SpaceFibre Frames in SpaceWire Packets

The application SpaceWire packet comprises a series of data characters followed by an end of packet marker (EOP). This is segmented into a number of segments, each of which is placed in a SpaceFibre frame with a start of data frame (SDF) added to the front of the segment and an end of data frame (EDF) at the end. Each SpaceFibre frame is then placed in a separate SpaceWire packet for sending over the SpaceWire link/network. A destination address is added at the front which contains the virtual channel number for the SpaceFibre frames and an EOP is added at the end.

Advantages:

- Uses SpaceWire for lower layer
- SpaceWire is proven in space flight applications
- Covers 1 Mbits/s to 300 Mbits/s speed range
- Compatible with existing SpaceWire devices

Disadvantages:

- Not galvanically isolated

10.4 SpaceFibre over SpaceWire with 8B/12B Data-Strobe Encoding

One of the most significant problems with SpaceWire is that its signals are not AC coupled, making galvanic isolation difficult. STAR-Dundee Ltd has devised a coding scheme that can be used with SpaceWire to transform the data-strobe signals into a form where they can be AC coupled. Each SpaceWire character is given a 12-bit code, which when serialised is DC balanced so that it can be AC coupled. Furthermore when encoded using data-strobe encoding the strobe signal is also DC balanced.

The main disadvantage is that the encoding is inefficient (33% coding overhead) compared to standard SpaceWire and 8B/10B encoding (both around 20% coding overhead). This is not a significant problem for slower data rates given that SpaceWire implementations are capable of raw data rates 300 Mbits/s in current radiation tolerant ASIC technology.

Note that STAR-Dundee Ltd has a patent application pending for 8B/12B data-strobe encoding.

Advantages:

- Galvanically isolated version of SpaceWire

Disadvantages:

- Reduced encoding efficiency
- Same cable mass as SpaceWire

10.5 SpaceFibre with 8B/12B Data-Strobe Encoding

One possible advantage of data-strobe encoding, which is not taken advantage of in SpaceWire is that it is possible to stop the clock, i.e. to freeze the data and strobe signals, at any time to save power. This has a limited effect with LVDS, but is significant when communicating over relatively short distances using LVTTL CMOS. The SpaceWire standard uses the exchange of silence on the line to signal an error and to start re-initialisation of the link. To take advantage of 8B/12B data-strobe encoding the link initialisation state machine of SpaceWire would need to be change. That of SpaceFibre is more appropriate. This leads to the concept of using SpaceFibre with the encoding and serialisation layers replaced by 8B/12B data-strobe encoding.

Advantages:

- Ability to stop clock to save power and instantly restart it
- Saves power when using LVTTL over short distances

Disadvantages:

- May require changes to SpaceFibre lane initialisation state machine
- Reduced encoding efficiency
- Same cable mass as SpaceWire

10.6 Trade Off of Alternative Solutions

A trade-off of the alternative solutions is provided in Table 10-1.

Table 10-1 Trade-Off of Alternative Solutions

Trade-Off	SpFi-LVDS	SpFi-OS-LVDS	SpFi over SpW	SpFi-SpW-8B12B	SpFi-8B12B
Speed Range	10-600 Mbits/s	1 to 100 Mbits/s	1 to 300 Mbits/s	1 to 200 Mbits/s	1 to 200 Mbits/s
Low Power	Yes	Yes	No	No	No
Low Mass	Yes	Yes	No	No	No
Qualified FPGA	No	Yes	Yes	Yes	Yes
PLL needed	PLL	No PLL	No PLL	No PLL	No PLL
Galvanic Isolation	Yes	Yes	No	Yes	Yes
Compatible with SpaceWire	No	No	Yes	No	No
Efficient coding	Yes	Yes	Yes	No	No
Significant modification	No	No	No	Yes	Yes

The cells in the table that have been shaded green show the acceptable features of each proposed solution.

Speed range: is an indication of the usable range of data signalling speeds that could be achieved. All of the potential solutions would cover the slower data rate required. The SpFi-LVDS approach may be restricted by the range of the PLL used.

Low power: is an indication of whether the link is likely to take less power than a SpaceWire link. The SpFi-LVDS and SpFi-OS-LVDS have reduced power consumption compared to the solutions using data-strobe encoding, because they use half the number of line drivers/receivers.

Low mass: indicates whether the link cable mass is likely to be significantly less than SpaceWire. The SpFi-LVDS and SpFi-OS-LVDS have much reduced cable mass compared to the implementations using data-strobe encoding, because they use half the number of signal wires.

Qualified FPGA: indicates whether it is possible to implement the solution in current space qualified FPGA technology, the key issue being the use of PLLs for clock recovery. The

SpFi-LVDS cannot be implemented in a current space qualified FPGA since it requires a PLL or similar clock recovery circuit. An external SerDes device would be required.

PLL needed: indicates whether PLL or similar clock recovery technology is required. The SpFi-LVDS needs a PLL.

Galvanic isolation: indicates whether the signals are DC balanced to support galvanic isolation. All solutions except SpaceFibre over SpaceWire can provide galvanic isolation.

Compatible with SpaceWire: indicates whether the solution is compatible with existing SpaceWire technology. Only the SpaceFibre over SpaceWire solution is compatible with existing SpaceWire interfaces.

Efficient coding: indicates whether the coding scheme is as efficient as 8B/10B encoding. The solutions that use 8B/12B encoding are not efficient from the signal coding point of view.

Significant modification: indicates whether the solution requires substantial modification to the SpaceWire or SpaceFibre layers that it has adopted. The solution that use 8B/12B encoding require significant modification to the SpaceFibre or SpaceWire standards.

Each of the potential solutions will now be considered in turn.

SpFi-LVDS: The main drawback of this approach is that it requires a PLL.

SpFi-OS-LVDS: This approach meets covers most of the trade-off points. The only issue that it does not cover is the compatibility with existing SpaceWire devices.

SpFi over SpW: This is higher mass and power than the SpFi-OS-LVDS approach, and does not provide galvanic isolation. It is, however, compatible with existing SpaceWire technology.

SpFi-SpW-8B12B: This approach has higher mass and power than SpFi-OS-LVDS. It provides galvanic isolation. It is not compatible with SpaceWire and requires significant modification to the SpaceWire standard.

SpFi-8B12B: This approach has higher mass and power than SpFi-OS-LVDS and does provide galvanic isolation. It is not compatible with SpaceWire and requires significant modification to the SpaceFibre standard.

From the outside interfaces SpFi-LVDS and SpFi-OS-LVDS are the same, the only difference being the implementation of the clock recovery circuit. For this reason they may be considered, from a standards perspective, to be one solution. This offers low mass, low power, galvanic isolation and can operate at moderate speeds in current space qualified FPGA technology, and at higher speeds with future FPGA technology.

The SpFi over SpW approach is attractive because of its backwards compatibility with SpaceWire.

The 8B/12B solutions will not be considered further.

11 Coherent Set of Protocols

Following the trade-off of the possible technologies for implementing slower speed SpaceFibre, the following solutions are selected for further research within the SpaceWire-RT project:

- SpaceFibre Fibre Optic
- SpaceFibre CML
- SpaceFibre LVDS (which includes SpFi-OS-LVDS)
- SpaceFibre over SpaceWire

These protocols are illustrated in the combined protocol stack of Figure 11-1.

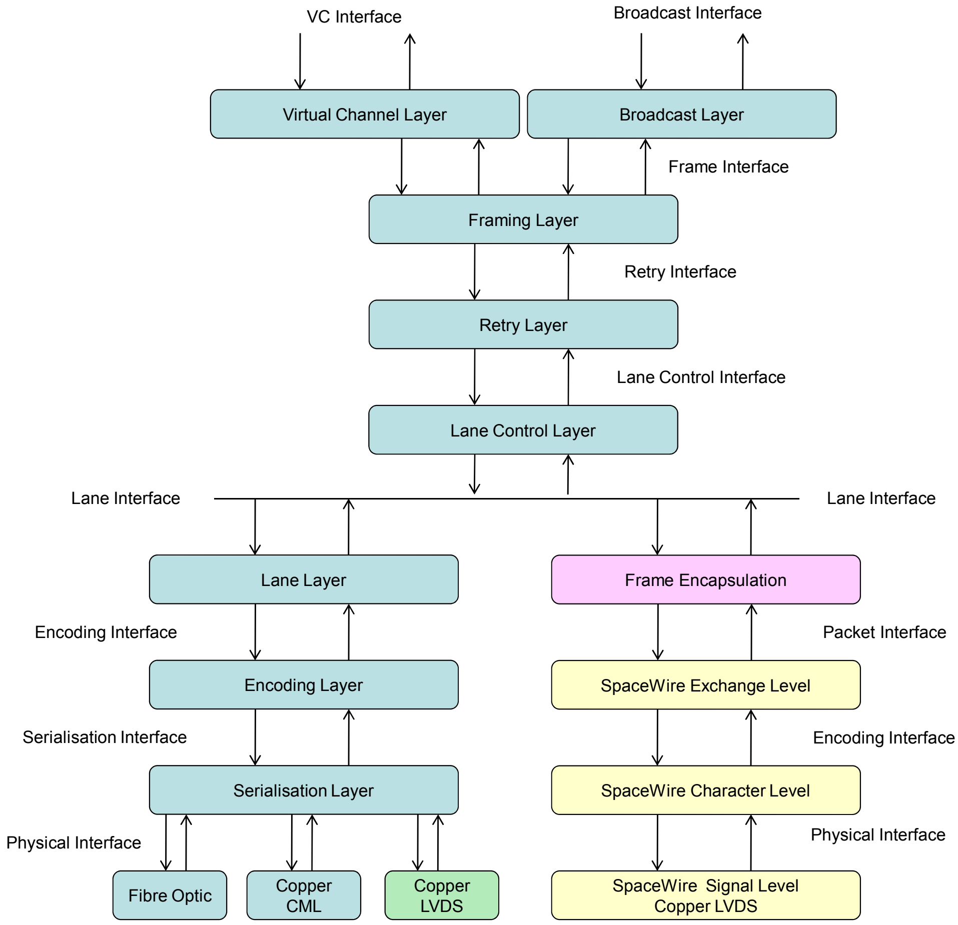

Figure 11-1 SpaceWire-RT Protocol Stack

The blue elements in the SpaceWire-RT protocol stack are the SpaceFibre layers. The green element is the LVDS physical layer for SpaceFibre-LVDS (including SpFi-OS-LVDS). The yellow elements are SpaceWire layers with the red element providing the SpaceFibre frame encapsulation function for running SpaceFibre over SpaceWire.

The key characteristics of the different SpaceWire-RT protocols are detailed in Figure 11-1.

Table 11-1 SpaceWire-RT Protocol Characteristics

Characteristic	SpFi-FO	SpFi-CML	SpFi- LVDS	SpFi over SpW
Media	Fibre Optic	Copper CML	Copper LVDS	Copper LVDS
Encoding	8B/10B	8B/10B	8B/10B	Data-Strobe
Speed Range	0.1 to 20 Gbits/s 50 Gbits/s in future	0.1 to 20 Gbits/s 50 Gbits/s in future	1 to 600 Mbits/s 1 to 50 Mbits/s OS	1 to 300 Mbits/s
Distance	100 m	5 m	10 m	10 m
Galvanic Isolation	Yes	Yes	Yes	No
Packet Size	Arbitrary	Arbitrary	Arbitrary	Arbitrary
SpaceWire Packet Level	Yes	Yes	Yes	Yes
Latency (TBC)	1 µs	1 µs	1 µs	10 1 µs
Cable Mass	< 30g/m	< 30g/m	< 30g/m	~87 g/m
Power (TBC)	< 200 mW	< 200 mW	< 200 mW	< 400 mW
QoS BW Reserved	Yes	Yes	Yes	Yes
QoS Priority	Yes	Yes	Yes	Yes
QoS Scheduled	Yes	Yes	Yes	Yes
QoS Best Effort	Yes	Yes	Yes	Yes
Broadcast Message	Yes	Yes	Yes	Yes
Determinism	Yes	Yes	Yes	Yes
Reliability	Yes	Yes	Yes	Yes

Fault Detection	Yes	Yes	Yes	Yes
Fault Isolation	Yes	Yes	Yes	Yes
Retry	Yes	Yes	Yes	Yes
SpaceWire compatible	No	No	No	Yes

12 SpaceFibre Network Layer

At present the SpaceFibre standard does not have a network layer defined, because effort has focused on the specification of the link layer protocol. SpaceFibre is intended to use the SpaceWire network level protocol. This section gives an overview of the SpaceFibre network layer and how virtual channels are used and supported at the network level.

12.1 SpaceFibre Applications

SpaceFibre is specifically designed for handling data on-board spacecraft. It can be used to provide point to point connections between equipment or using SpaceFibre routers to provide a complete interconnection network.

An example spacecraft data handling architecture using SpaceFibre point to point links is illustrated in Figure 12-1.

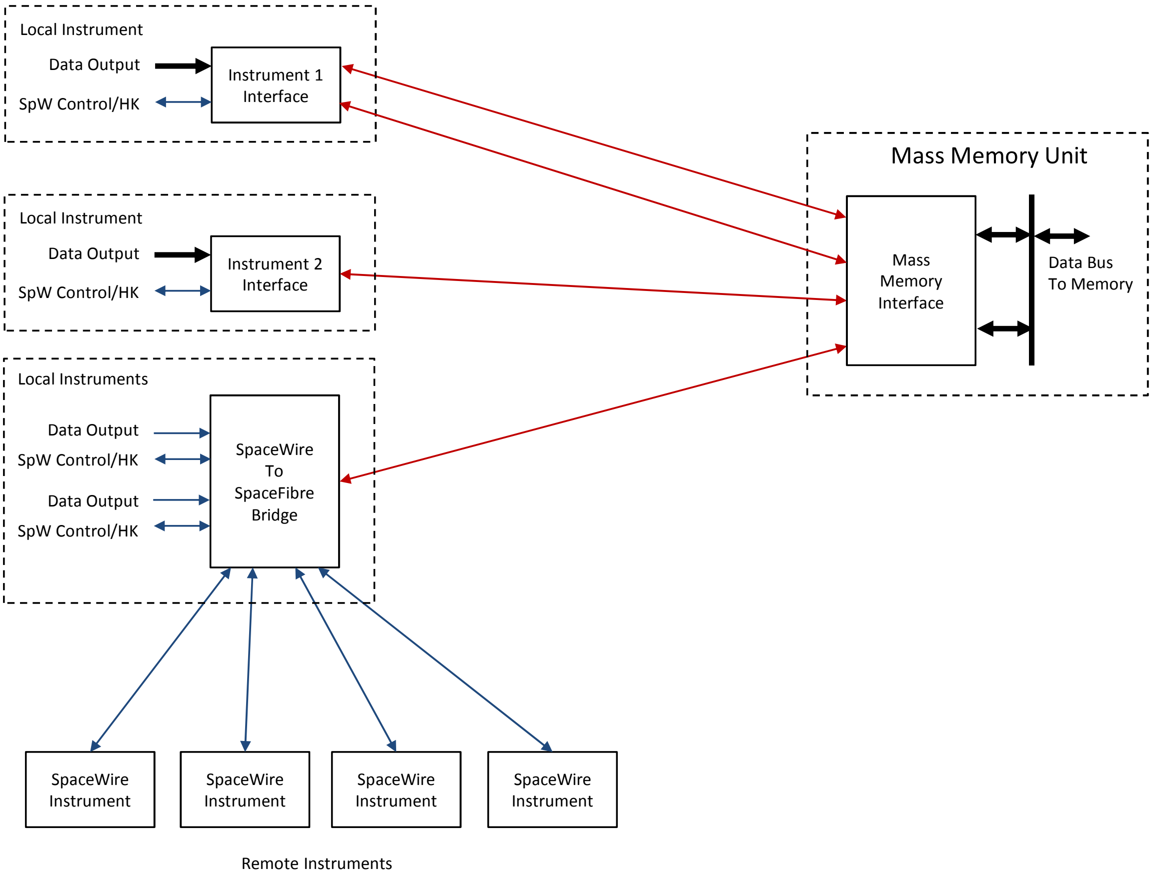

Figure 12-1 Spacecraft Data Handling with SpaceFibre Point to Point Links

SpaceFibre is being used to connect various instruments to the on-board mass memory unit. Instrument 1 is a very high data-rate instrument which requires two SpaceFibre links operating together to transfer data from the instrument to the mass memory. Instrument 2 is a high data-rate instrument with a single SpaceFibre link connecting it to the mass memory. There are several moderate data-rate instruments each with a SpaceWire interface. Rather than connect all of these SpaceWire links directly to the mass memory unit, a SpaceWire to SpaceFibre Bridge device is used which is able to multiplex several SpaceWire links over a single SpaceFibre link that goes to the mass memory unit. This single SpaceFibre link saves significant mass on the spacecraft if it has to run over a few metres, compared to running the separate SpaceWire links.

The SpaceWire to SpaceFibre Bridge provides a separate virtual channel for each SpaceWire link. The quality of service for each of these virtual channels can be specified according to the needs of the individual instruments. For example, each virtual channel can be allocated a bandwidth corresponding to the expected bandwidth from the corresponding instrument.

When SpaceFibre is being used to provide point to point connections there is no need for a network layer. The packets being sent are SpaceWire packets, but there are no routers.

An example spacecraft data handling architecture that uses SpaceFibre routers is illustrated in Figure 12-2.

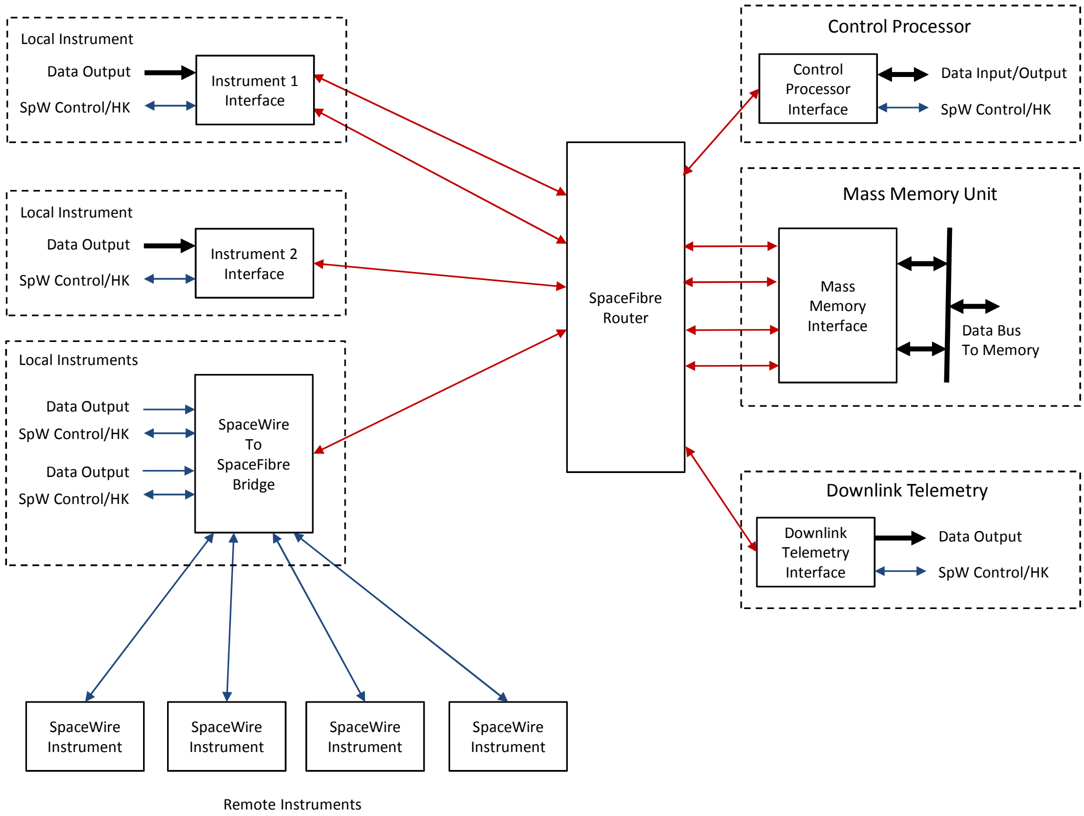

Figure 12-2 Spacecraft Data Handling with SpaceFibre Network

The data handling architecture of Figure 12-2 is similar to that of Figure 12-1, except that a control processor has been added that is used to configure and control all of the on-board data-handling equipment. It therefore needs a connection to every instrument and to the mass memory. This could be provided using a separate command and control network, but this would result in additional mass and power consumption. The addition of a SpaceFibre router connected to all of the on-board equipment allows the control processor to send commands and receive information from all of the on-board data handling units. No additional network is required. A small amount of network bandwidth can be reserved for the control processor, so that it can operate without regard to other traffic flowing over the network.

12.2 SpaceFibre Router

A SpaceFibre router with four SpaceFibre ports and an internal configuration port is illustrated in Figure 12-3.

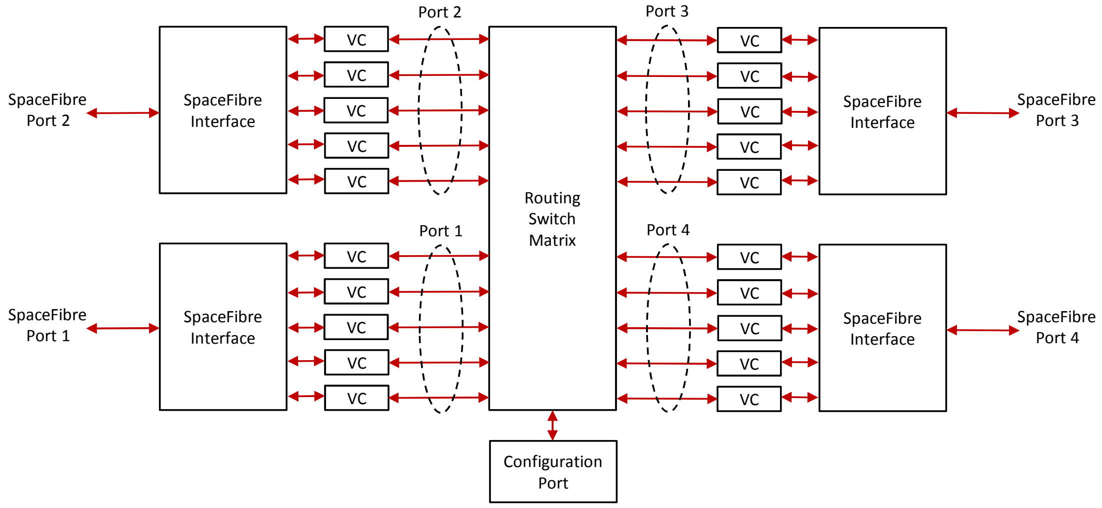

Figure 12-3 SpaceFibre Router

Each SpaceFibre port, in this example, has five virtual channels. The virtual channel number of each virtual channel buffer can be configured. The virtual channels from each SpaceFibre port are connected to a routing switch matrix. The routing switch switches SpaceWire packets being carried over SpaceFibre according to the leading address of the SpaceWire packet. It is able to switch several packets arriving on the same SpaceFibre port over different virtual channels at the same time, so that a packet flowing in one virtual channel does not affect that flowing in another virtual channel. The routing switch supports both path and logical addresses, as for a SpaceWire router.

A packet arriving at a SpaceFibre port on a certain number virtual channel can only be routed to a port that has a virtual channel with the same number. For example, a SpaceWire packet with path address 4, arriving at SpaceFibre port 2 on virtual channel 8, can only be routed to port 4 (as requested by the path address), if port 4 has one of its virtual channels configured to be virtual channel number 8.

12.3 SpaceFibre Virtual Networks

A SpaceWire network comprises nodes, links and routers. Nodes are the sources and destinations of SpaceWire packets, routers route packets towards their destinations, links connect nodes and routers to form the network. When a packet arrives at a router the leading data character of the packet determines what output port it is to be forwarded through. If this data character has a value of 0 to 31 it is a path address which directly references the output port number, e.g. a value of 4 will result in the packet being routed through port 4. If the leading data character has a value between 32 and 254 it is a logical address which is used to look up the output port in a routing table. The value 255 is a reserved value. Using path addressing a source is able to specify the path that a packet is to take through the network. In logical addressing the paths through the network are specified by the routing tables in the router. The logical address put on the front of a packet selects which one of these predetermined paths is to be used.

If two SpaceWire packets arrive at a router and are to be forwarded through the same output port, there is contention because only one packet can travel down a link at a time. The two packets compete for access to the output port of the router. The one that is granted access is forwarded through the output port and the other one has to wait. Because of the wormhole routing mechanism used in SpaceWire, where no packet buffers are used in the input and output ports of the router, the tail of a packet can be strung out across a path through the network causing other packets that want to use any of the links in that path to be delayed. Packets being sent by more than one source over a network may compete for the use of links in the network, i.e. compete for the network resources.

SpaceFibre is able to support multiple virtual networks on a single physical network. Packets sent by various sources using a particular virtual network will compete for the resources of that virtual network, just like packets compete in a SpaceWire network. Packets travelling over different virtual networks do not compete for network resources. Each virtual network is like a separate SpaceWire network. When several virtual networks run over the same physical SpaceFibre link frames from packets travelling over each of those links are multiplexed over the physical SpaceFibre link according to the quality of service specified for each virtual network. For example, bandwidth reservation quality of service can be used to allocate specific portions of the link bandwidth to each of the virtual networks using that link.

A virtual network is illustrated in Figure 12-4.

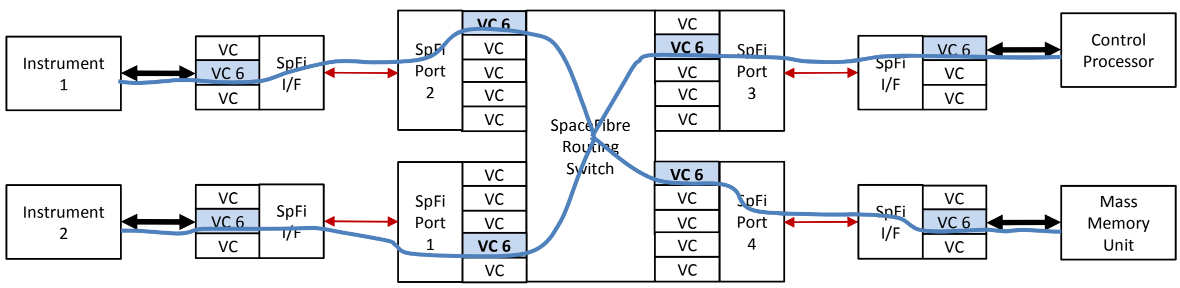

Figure 12-4 SpaceFibre Virtual Network

The virtual network illustrated is using virtual channel 6. Each SpaceFibre interface in the network has one of its virtual channels set to virtual channel number 6 (VC 6). A packet to be sent from the control processor to instrument 1 over this virtual network will be given path address 2 and be written into VC 6 of the SpaceFibre interface attached to the control processor. It will travel from the control processor over the SpaceFibre link to port 3 of the SpaceFibre router where the path address will cause it to be routed to port 2. The routing switch always transfers a packet to the same number VC as it arrives on, so it is transferred to port 2, VC6. The packet then travels across the SpaceFibre link to arrive in VC6 at instrument 1. Using VC6, the control processor can send and receive packets from any of the nodes on the network. The virtual network is ideal when one unit, e.g. a control processor, has to send and receive information from many other devices.

12.4 SpaceFibre Virtual Point to Point Connections

If both instruments were to send data to the mass memory over VC 6, they would compete for access to port 4 of the SpaceFibre router. Each of them would need to be allocated a separate virtual network. The instruments would then be able to send packets to any other node on the network. This is an excessive use of network resources and may not be desirable i.e. it may be useful to prevent instrument 1 sending packets to instrument 2 when this should never normally happen. To support data transfer over a SpaceFibre network from one specific node to another specific node, e.g. from instrument 1 to the mass memory unit, a virtual point to point connection is required. A virtual point to point connection is illustrated in Figure 12-5.

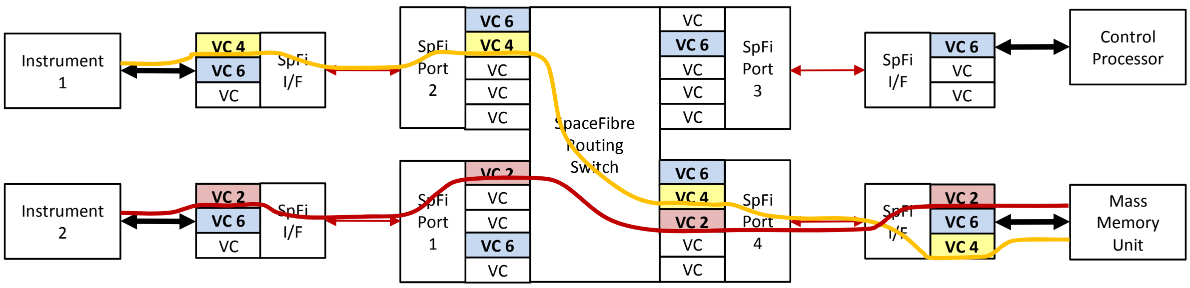

Figure 12-5 SpaceFibre Virtual Point to Point Connection

Instrument 1 and instrument 2 both want to send packets to the mass memory unit. This has to be done without one instrument inhibiting or interrupting the flow of packets from the other instrument. Instrument 1 uses virtual channel 4 and instrument 2 uses virtual channel number 2. The SpaceFibre router is configured to have VC2 in port 1, VC4 in port 2 and both VC2 and VC4 in port 4.

A packet to be sent from instrument 1 to the mass memory unit will be written into VC 4 and will have path address 4. It will travel over the SpaceFibre link to port 2 of the SpaceFibre router where the path address will cause it to be routed to port 4. The packet will be transferred to VC4 of port 4 and then across the SpaceFibre link to VC4 at the mass memory unit. Similarly instrument 2 sends its data to the mass memory using VC2 and path address 4.

Should instrument 1 inadvertently get the address of the packet wrong, e.g. send a packet with path address 3 over VC4, it will be discarded by the router because there is no VC4 in port 3. This gives an additional measure of fault isolation in a SpaceFibre network. Virtual point to point (VP2P) connections can be set up in the network and if the path through the SpaceFibre network defined by the path address does not match that defined by the virtual channel assignment, the packet will be discarded.

Another possibility is for the router configuration to define the output port that a packet arriving on a specific port and VC is to be sent to. This would in effect provide a virtual circuit between the source and the destination. For spacecraft applications the fault detection and isolation characteristics of the VP2P approach are considered to be beneficial.

Clearly a virtual point to point connection is just a restricted version of a virtual network and it is possible to have something in between, for example a virtual point to point connection that can send packets to one of two possible nodes. The main advantage of the VP2P concept compared to using many full virtual networks is that it uses fewer resources and allows better fault detection and isolation.

Virtual channel 0 is always a virtual network which is used for configuring the SpaceFibre network and which may also be used for configuring devices attached to the network.

12.5 Example SpaceFibre Application

A simple example of a SpaceFibre application is given in Figure 12-6. This combines the virtual network of Figure 12-4 with the virtual point to point connections of Figure 12-5.

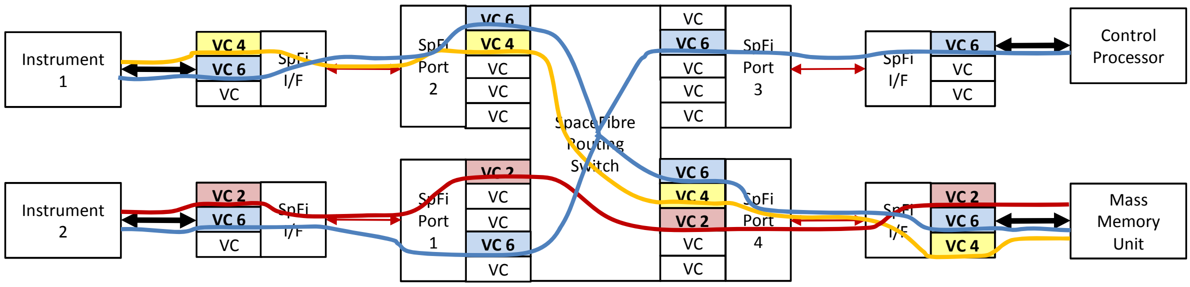

Figure 12-6 Simple SpaceFibre Application

The control processor can configure and control the two instruments and the mass memory unit and collect housekeeping information from them. Instrument 1 and instrument 2 are able to send data to the mass memory unit. The three virtual channels used are all independent with channel characteristics determined by the quality of service parameters. There is no packet blocking within the SpaceFibre network.

A more comprehensive spacecraft data handling network is illustrated in Figure 12-7 and Figure 12-8.

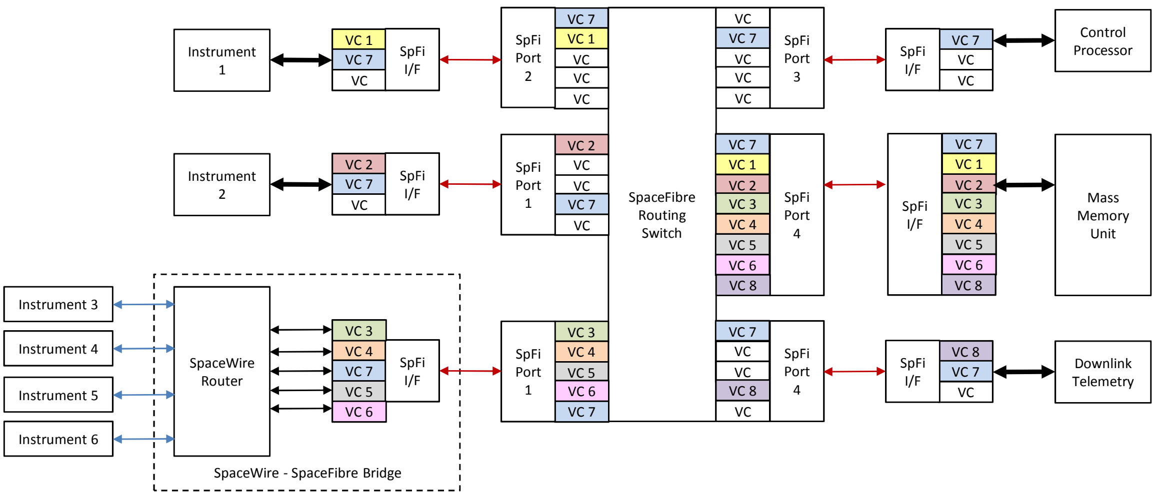

Figure 12-7 SpaceFibre Spacecraft Data Handling Application

The data handling system is similar to that of Figure 12-2, simplified a little to aid clarity. It comprises two high data-rate instruments with SpaceFibre interfaces and four instruments with SpaceWire interfaces. A mass memory unit stores data from the instruments and when appropriate sends the data to a downlink telemetry unit for transmission to ground. A control processor is used for configuring the instruments, mass memory and downlink telemetry unit and for reading housekeeping information from these units. A SpaceFibre routing switch is used to interconnect the various units. There is one virtual network which is used by the control processor to access all the other units. This uses VC7 which is shown in blue. Each unit and each port of the router has a virtual channel set to VC7. Instrument 1 and instrument 2 send data directly to the mass memory unit using VC1 and VC2 respectively. The mass memory sends data to the downlink telemetry unit using VC8.

The SpaceWire instruments are each connected to a SpaceWire to SpaceFibre bridge which incorporates a SpaceWire router. On the SpaceFibre side of this bridge there is one virtual channel for each SpaceWire instrument, plus one additional virtual channel (VC 7) to support the configuration, control and housekeeping virtual network. Each virtual channel is connected to a separate port on the SpaceWire router. To send over virtual channel 3 instrument 3 adds a path address to the packet to route it through the port of the router attached to port 3. When responding to a command from the control processor it routes the reply to the port attached to VC7, the virtual network. Since each SpaceWire instrument has a separate virtual point to point connection to the mass memory they do not interfere with one another, each being allocated an appropriate amount of network bandwidth.

There is potential contention on the SpaceWire links because they have to carry different types of information: instrument data and replies to commands. It is the responsibility of the SpaceWire instrument to interleave data packets and reply packets. Once on the SpaceFibre network they travel over different virtual channels. The command process may have to wait for a significant amount of time for a reply to a command sent to one of the SpaceWire instruments. Because of this it should send posted commands, rather than waiting for the reply to a command before sending the next one.

The virtual network and virtual point to point connections are highlighted in Figure 12-8.

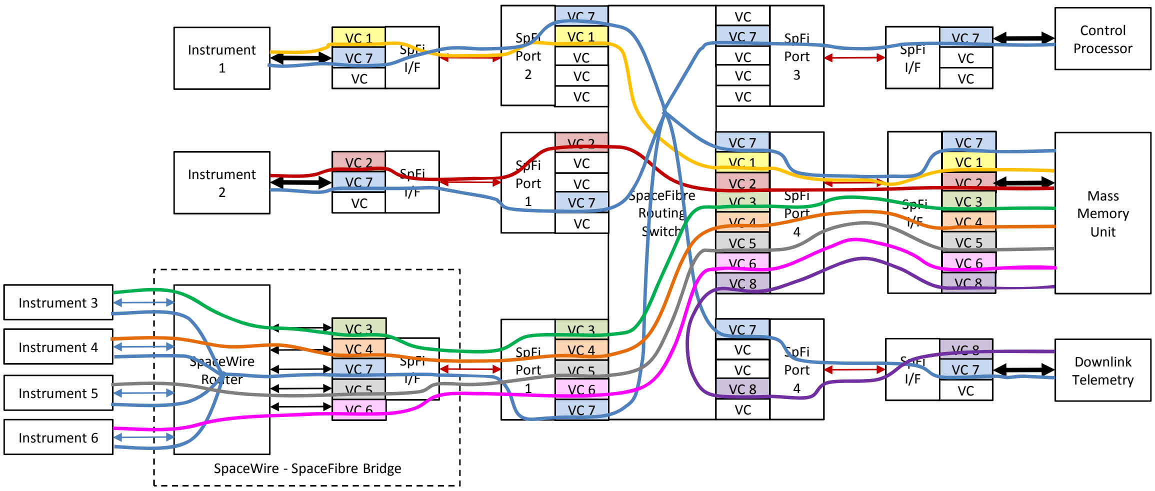

Figure 12-8 SpaceFibre Spacecraft Data Handling Application with Virtual Network and Virtual Point to Point Connections Highlighted

13 Outline Specification

This section provides an outline specification of the proposed SpaceWire-RT protocols.

13.1 Applicable Documents

AD1 European Cooperation for Space Standardization, Standard, “SpaceWire, Links, Nodes, Routers and Networks”, Issue 1, European Cooperation for Space Data Standardization, 31 July 2008.

AD2 Parkes SM, Ferrer Florit A, Gonzalez A, and McClements C, “SpaceFibre”, Draft D, Space Technology Centre, University of Dundee, 29th February 2012.

13.2 SpaceWire-RT Protocol Stack

a) The SpaceWire-RT protocol stack shall be as illustrated in Figure 13-1.

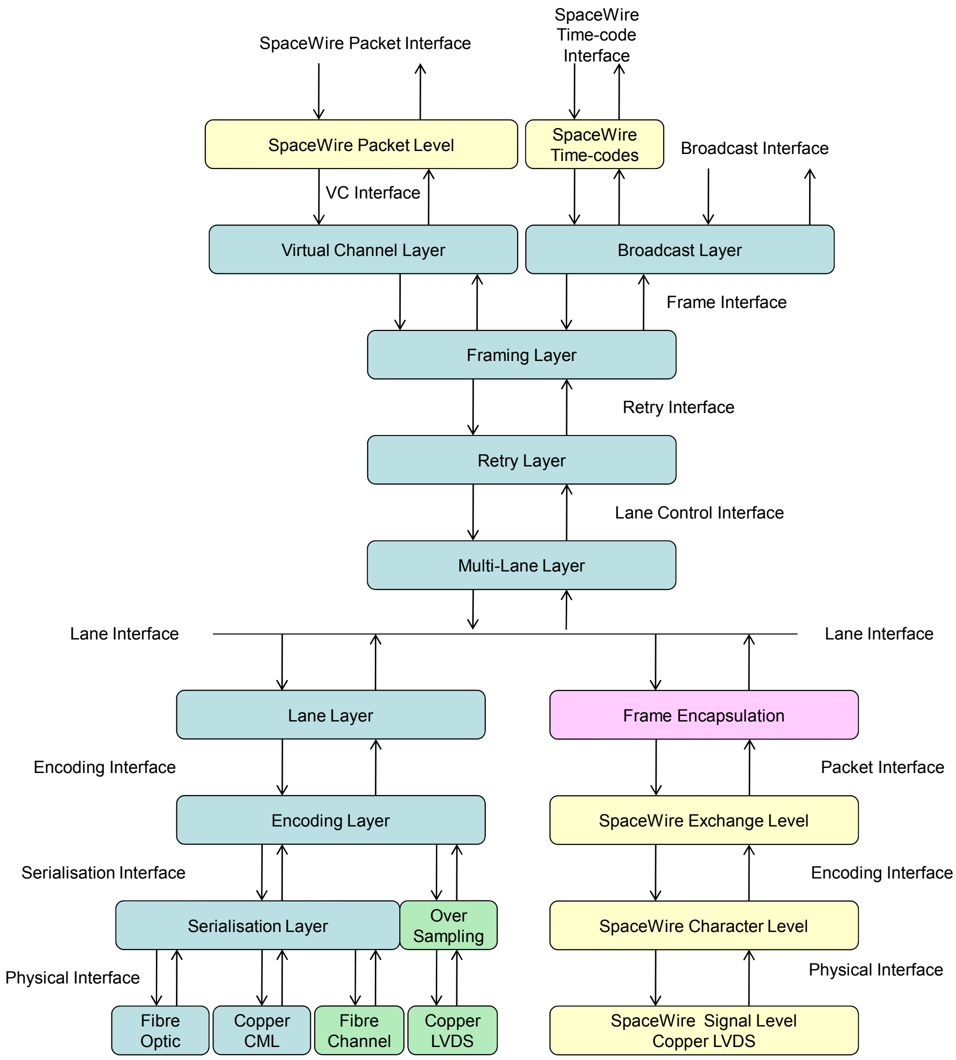

Figure 13-1 SpaceWire-RT Protocol Stack Showing Different Serialisation and Physical Layer Options

Note that the terms level and layer are both used to refer to layers of the protocol stack. The term level is used when a layer of the SpaceWire protocol stack is being referred to (level is the term used in the SpaceWire standard and also in the IEEE1355-1995 standard). The term layer is used for all other layers.

13.3 SpaceWire-RT Application Interfaces

a) There shall be three application interfaces to the SpaceWire-RT protocol stack:

i. SpaceWire Packet Interface

ii. SpaceWire Time-code Interface

iii. Broadcast Message Interface

b) The SpaceWire Packet Interface shall be responsible for sending and receiving SpaceWire packets over a specific virtual channel.

c) The SpaceWire Time-code Interface shall be responsible for sending and receiving SpaceWire time-codes over the SpaceWire-RT network.

d) The Broadcast Message Interface shall be responsible for sending and receiving broadcast messages over the SpaceWire-RT network.

13.4 SpaceWire-RT Packets

a) The SpaceWire Packet Layer shall follow the SpaceWire packet format defined in the SpaceWire standard [AD1].

13.5 SpaceWire-RT QoS

a) The Virtual Channel, Framing and Retry Layers shall jointly be responsible for providing quality of service for SpaceWire-RT.

b) The Virtual Channel Layer shall follow the Virtual Channel Layer specification provided in the SpaceFibre standard [AD2].

c) The Framing Layer shall follow the Virtual Channel Layer specification provided in the SpaceFibre standard [AD2].

d) The Retry Layer shall follow the Virtual Channel Layer specification provided in the SpaceFibre standard [AD2].

13.6 SpaceWire-RT Multi-Laning

a) The Multi-Laning layer shall follow the Laning Layer specification provided in the SpaceFibre standard [AD2].

13.7 SpaceWire-RT SpaceFibre Lane

a) The Lane Layer shall follow the Laning Layer specification provided in the SpaceFibre standard [AD2].

b) The Encoding Layer shall follow the Encoding Layer specification provided in the SpaceFibre standard [AD2].

c) The Serialisation Layer shall follow the Serialisation Layer specification provided in the SpaceFibre standard [AD2].

d) The Physical Layer shall follow the Physical Layer specification provided in the SpaceFibre standard [AD2], including both Fibre Optic and Copper media options.

13.8 SpaceWire-RT SpaceFibre with LVDS

a) The Serialisation Layer shall permit the use of oversampling to perform bit synchronisation in the receiver.

   Note: this will significantly reduce the bit rate but will enable implementation without a phase locked loop or similar clock recovery technology.

b) The Physical Layer shall provide an additional option to use LVDS instead of CML running over copper cable.

c) The Physical Layer shall provide an additional option to use Fibre Channel physical layer instead of CML running over copper cable.

13.9 SpaceWire-RT over SpaceWire

a) A Frame Encapsulation Layer shall be provided which encapsulates SpaceFibre frames in SpaceWire packets so that the frames can be sent over a SpaceWire network.

b) When running over SpaceWire each SpaceFibre Frame shall be encapsulated in a SpaceWire packet, as illustrated in Figure 13-2.

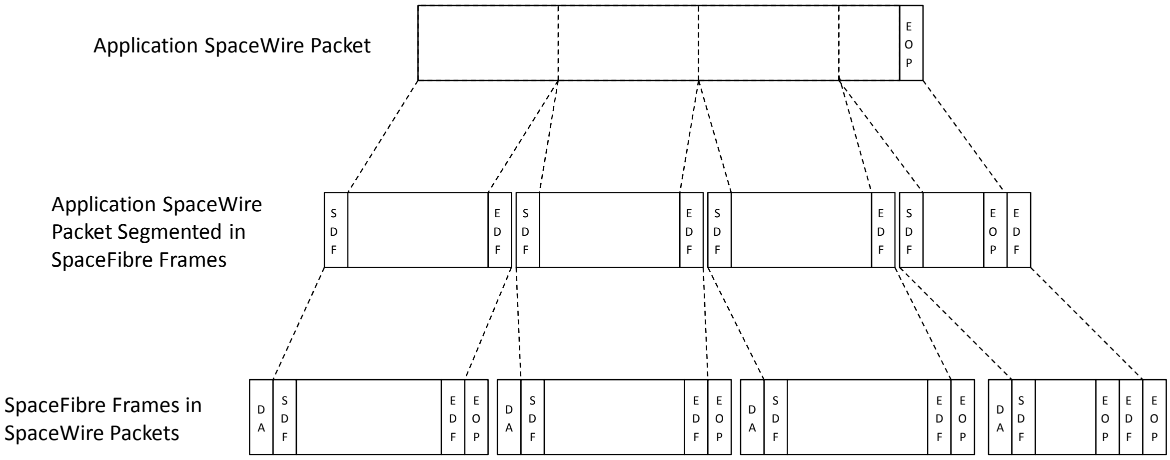

Figure 13-2 Frame Encapsulation in a SpaceWire Packet

c) The destination address of the packet that the Frame is part of shall be added to the start of the packet containing a Frame.

d) The cargo of the packet shall contain the complete frame including the start of frame word and end of frame word.

e) The SpaceWire packet shall be terminated with an End of Packet Marker.

f) The SpaceWire packet shall contain one frame only.

13.10 SpaceWire-RT SpaceWire Exchange Layer

a) The SpaceWire Exchange Layer shall follow the SpaceWire Exchange Level defined in the SpaceWire standard [AD1].

13.11 SpaceWire-RT SpaceWire Character Layer

a) The SpaceWire Character Layer shall follow the SpaceWire Character Level defined in the SpaceWire standard [AD1].

13.12 SpaceWire-RT SpaceWire Signal Layer

a) The SpaceWire Signal Layer shall follow the SpaceWire Signal Level defined in the SpaceWire standard [AD1].

13.13 SpaceWire-RT Broadcast Messages

a) The Broadcast Message Layer shall follow the Broadcast Message Layer specification provided in the SpaceFibre standard [AD2].

13.14 Sending Time-Codes as Broadcast Messages

a) Time-codes shall be transmitter over a SpaceWire-RT network encapsulated in a broadcast message.

Note: the way in which time-codes are encapsulated in broadcast messages has yet to be defined. One example is to place the time-code time and flags fields into the reserved field of time type of broadcast message. Another possibility is to use a distinct type of broadcast message to carry time-codes.

13.15 Oversampling Serialisation Layer

a) SpaceWire-RT shall permit recovery of the received data stream using oversampling as well as phase-locked loop clock recovery techniques.

b) The two ends of the link shall operate at the same bit rate with a maximum permitted difference in bit clocks between the two ends of the link of 1% (TBC).

c) The receiver bit synchronisation circuitry shall track any change in the receive bit interval and sample the received data bit within +/- 25% of the centre of the bit interval.

d) The received data shall be sampled and de-serialised and passed to the encoding layer for decoding.

13.16 SpaceFibre LVDS

a) SpaceWire-RT shall permit the use of an LVDS physical layer with SpaceFibre.

Note: LVDS is not capable of the Gbits/s signalling speed of CML.

13.17 SpaceFibre Fibre Channel Physical

a) SpaceWire-RT shall permit the use of a Fibre Channel type of physical layer.

Note: This is to provide relatively long distance communication (30 m) at data rates of up to 1 Gbits/s.

14 Conclusion

A coherent set of communication protocols have been defined that covers most of the applications for serial data link and network technology on board spacecraft, including payload data-handling and avionic applications. A literature survey on real-time network concepts has been presented. The requirements from WP1 have been analysed and the key challenges for SpaceWire-RT identified. Based on these requirements and key challenges a set of evaluation criteria have been synthesised. Research on QoS mechanisms suitable for use with SpaceFibre has been carried out, resulting in the design of a simple, powerful, and comprehensive quality of service mechanism. This QoS mechanism has then been extended to include specific classes of fault detection in support of FDIR. SpaceFibre has been evaluated against the SpaceWire-RT requirements and its shortcomings identified. Potential solutions that cover those shortcomings have been proposed and traded-off using the evaluation criteria, resulting in a coherent set of protocols. An outline specification for these protocols has then been provided, building on existing SpaceFibre and SpaceWire protocols as far as possible, and extending them where necessary.

The QoS mechanisms developed in the SpaceWire-RT project have been adopted into the SpaceFibre standard specification and presented to the SpaceWire working group.

All the tasks of WP2 have been completed and reported in this document. The objectives of WP2 have been achieved.

The next steps in the SpaceWire-RT project are WP3, WP4 and WP5 which will run in parallel.

- WP3, SpaceWire-RT Validation and Simulation, will validate the QoS, FDIR and some other layers of SpaceFibre through simulation. The results of the simulation will be used to update the SpaceWire-RT specification and to inform the VHDL IP Core Development (WP4) and ASIC Feasibility and Prototyping (WP5) activities.
- WP4, VHDL IP Core Development, will explore “oversampling” and “SpaceFibre over SpaceWire”, implementing them in the form of IP cores written in VHDL.
- WP5, ASIC Feasibility and Prototyping, will investigate ASIC technologies appropriate for implementation of SpaceWire-RT. Initial design and core prototyping activities will be undertaken, to ensure that the principal risk areas with an ASIC development have been addressed.