SpaceWire Internet Tunnel

S. Mills[1], S. M. Parkes[1] and R. Vitulli[2]

[1]*University of Dundee, Applied Computing, Dundee, DD1 4HN, Scotland, UK.*
smills@computing.dundee.ac.uk, sparkes@computing.dundee.ac.uk .

[2]*European Space Agency, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands*
Raffaele.Vitulli@esa.int

Abstract

The current methods of building spacecraft often involve partners spread across several countries developing sub-systems separately. Only once these sub-systems are complete are they brought together and integration testing performed. Faults identified at this late stage of development can be very costly.

This paper describes the SpaceWire Internet Tunnel, a means of connecting SpaceWire sub-systems which are geographically separated. Using this Tunnel, integration testing of SpaceWire systems may be performed at an earlier stage, drastically reducing the effects of any faults identified. The advantages of using a SpaceWire Internet Tunnel are listed, the hardware and software of the Tunnel is described and the results of tests performed on the Tunnel are presented.

1. Introduction

Spacecraft are often built by consortia with partners spread across several countries. Development teams are separated geographically and to some extent by culture and language. During spacecraft design, emphasis is on using documentation to help ensure the correct operation of the system. Detailed interface specifications are written for each subsystem and agreed between prime contractor and subcontractor. Misunderstandings or misinterpretations of specification lead to contract change notes and often to increased costs. Once the sub-systems have been developed, integration and testing requires the movement of the various units to a common location. This is usually done late in the development programme. Any fault or error in specification or design found at this late stage is expensive causing delays in the integration and test programme and may delay the launch of the spacecraft. Risk to the project is increased by the integration of the components late in the project schedule.

Standardization of interfaces reduces the risk of late integration significantly. At least the physical and data link parts of the interconnection should work without any difficulty. SpaceWire [1] [2] is one onboard communications standard that defines the physical and data link layers of an interconnection. SpaceWire is designed to connect high data-rate sensors, large solid-state memories, processing units and the downlink telemetry

subsystem providing an integrated onboard, data-handling network. For European space missions a proven SpaceWire interface design is available from ESA in VHDL form [3], simplifying the development of new SpaceWire devices and helping to ensure their compatibility. For non space applications and non European space applications this VHDL code is available from the University of Dundee. Standardization of higher level protocols for SpaceWire is currently underway [4]. While these standards will help reduce the problems of late system integration, the applications running over the communication still have to be developed and tested in isolation, prior to integration. A method is needed to enable earlier integration of the spacecraft electronic sub-systems while they are still geographically separated.

Virtual satellite integration [5] is a means of interconnecting the sub-systems early on in the development life cycle without bringing them to a centralised integration and test facility. Connection is provided by the Internet. Early interconnection reduces risk due to misunderstanding or misinterpretation of equipment specifications, resulting in a more manageable project.

2. SpaceWire Virtual Integration

Figure 1 shows a SpaceWire network that is separated into two distinct subsystems. For example, sub-system 1 centred around router 1 could be an instrument and sub-system 2 could be a centralized data-handling unit. These sub-systems may well be developed by separate organizations in different countries. To test the two sub-systems together requires bringing them to a common location and running tests there. This is normally done after development of the two units is complete and any problems found during integration are normally expensive to resolve and will delay the overall project. Significant effort is expended early on in the project to clearly define the physical and functional interfaces between the two units to try to avoid any problems during integration.

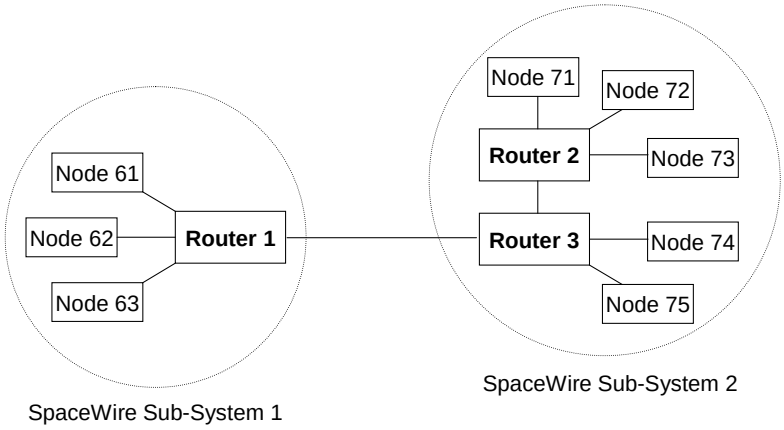

Figure 1: SpaceWire Network Comprising Two Distinct Sub-Systems

The SpaceWire standard is one step towards a solution to this “late integration” problem. Using a common standard for the physical connections between the two units will ensure that they can make a connection and exchange packets of data successfully.

Once units can exchange data the next step is to make sure that the contents of the packets and the order in which packets are exchanged is appropriate and meaningful to each sub-system. This is the functional testing of the complete system and is where a SpaceWire IP Tunnel can help.

Figure 2 shows the two subsystems separated geographically. The SpaceWire connection that joined the two units has been replaced by a tunnel though the Internet. The SpaceWire link from sub-system 1 is connected to a PC which is in turn connected to the internet. SpaceWire packets arriving at the PC from sub-system 1 are transferred across the internet to the PC attached to sub-system 2. Here the SpaceWire packet is reconstructed and sent across the SpaceWire link to router 3.

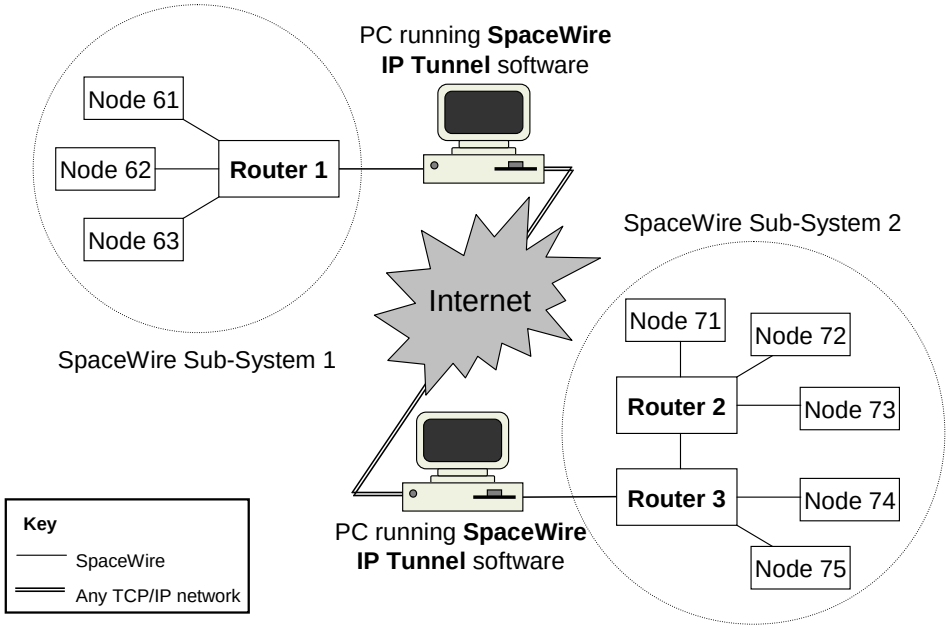

Figure 2: SpaceWire IP Tunnel

The SpaceWire link between router 1 on SpaceWire sub-system 1 and router 3 on SpaceWire sub-system 2 has been replaced by a tunnel through the Internet. The two routers will be able to communicate as if they were connected using SpaceWire, with only the time taken for the packets to be delivered providing any indication that the routers are not directly connected. This means that the two sub-networks can operate as one integrated network, and nodes such as 62 will be able to communicate with nodes 73 and 74, for example, as if they were on the same network.

Functional testing of the complete system can now take place before they are integrated physically. This “virtual integration” allows a significant amount of testing to be completed prior to final physical integration. Many problems can be resolved early in the development life cycle reducing the cost of fixing them. Due to the bandwidth limitations and long latency of the interconnection it is not possible to perform the communication in real-time, but a significant amount of functionality can be tested never the less.

The SpaceWire IP Tunnel may also be used to connect remote simulations (e.g. spacecraft, environment or observation target simulations) into the network, further helping with testing and debugging complex data-handling sub-systems.

3. SpaceWire IP Tunnel

The application domain model for the SpaceWire IP Tunnel software is shown in Figure 3.

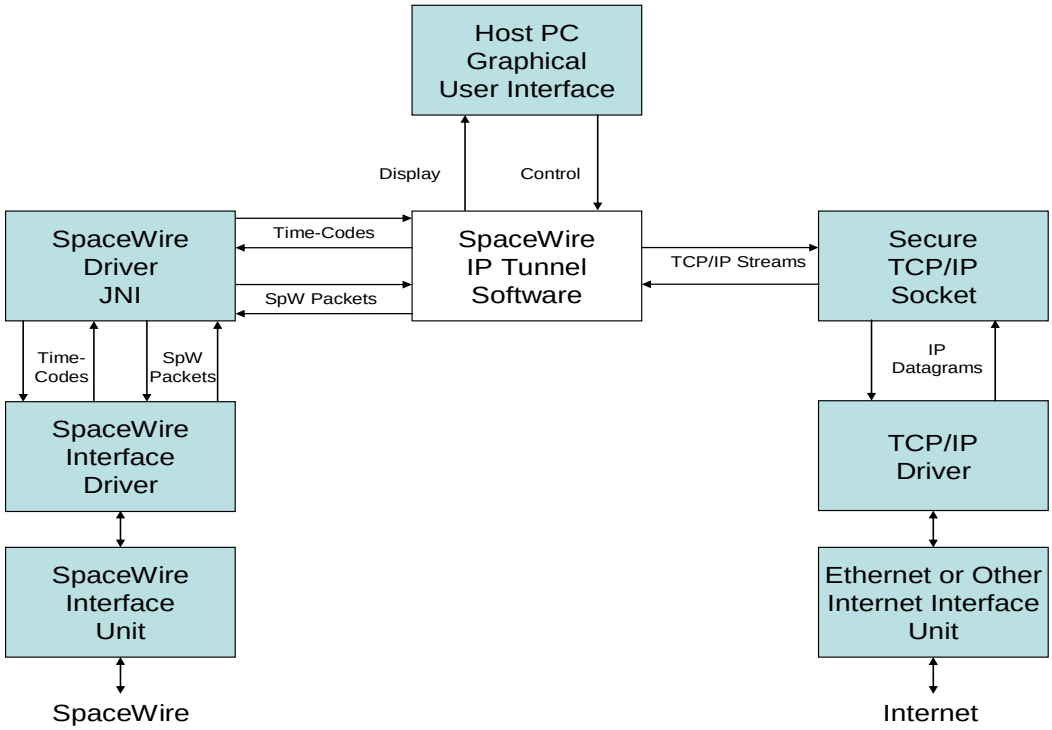

Figure 3: SpaceWire IP Tunnel Application Domain Model

The SpaceWire IP Tunnel talks to three other sub-systems: the SpaceWire interface, the internet interface and the GUI of the host PC.

The SpaceWire interface provides SpaceWire packets and time-codes to be transferred across the Tunnel and receives them from the Tunnel to be sent out over the SpaceWire interface. The SpaceWire interface comprises a SpaceWire interface unit, an appropriate driver for the SpaceWire interface unit and a Java Native Interface (JNI) to enable the SpaceWire IP Tunnel software to talk to the SpaceWire interface driver.

The internet interface sends and receives SpaceWire packets and time-codes encapsulated in a secure TCP/IP datagram stream. Any form of internet interface (typically Ethernet) can be used provided that there is a TCP/IP driver available for the interface. The secure internet connection is the actual IP Tunnel. The SpaceWire IP Tunnel software provides the two-way entrances and exits to the IP Tunnel.

SpaceWire packets arriving over an attached SpaceWire link are received by the SpaceWire interface unit and driver and passed through the SpaceWire Driver JNI to the SpaceWire IP Tunnel software. This software passes the packets and time-codes to the

Secure TCP/IP socket which transfers them via a TCP/IP driver and an Ethernet or other internet interface out to the internet. When the packets or time-codes arrive at the other end of the SpaceWire IP Tunnel they are received by the Ethernet or other internet interface unit and passed via the TCP/IP driver through the Secure TCP/IP socket interface to the SpaceWire IP Tunnel software. The SpaceWire IP Tunnel software checks the received packets and time-codes and passes them on through the JNI interface to the SpaceWire driver. Finally they are sent by the SpaceWire driver out of the SpaceWire interface unit.

A man-machine interface is needed for the SpaceWire IP Tunnel to configure the various Tunnel parameters, to provide operator control of the Tunnel and to report any errors and the status of the Tunnel. The man-machine interface is part of the SpaceWire IP Tunnel software and is supported by the GUI of the host PC. Any errors during transfer are reported to the user via the man-machine interface of the SpaceWire IP Tunnel software which uses the GUI of the host PC to display error messages.

4. Tunnel Hardware

The SpaceWire IP Tunnel software can be used with existing SpaceWire devices available from STAR-Dundee, such as the SpaceWire Router-USB and SpaceWire-USB Brick [6]. When used with these devices the Tunnel has some limitations. For example, standard SpaceWire interfaces do not maintain the order of time-codes with data packets as time-codes should be treated with higher priority.

In order to provide the full functionality required for a SpaceWire Tunnel, a modified device was required. This device, the SpaceWire Tunnel device [6], maintains the order of data characters and time-codes, along with links disconnecting and starting. It also provides other features required to tunnel traffic, such as the ability to send time-codes over a specific link.

5. Tunnel Software

In order to support these hardware modifications, changes were required to the driver which provides the interface with the device. For example, functionality to send chunks of packets, as opposed to full packets, was added to allow time-codes to be interleaved at the correct position within data packets. Performance of the driver was also improved, to allow these small traffic items to be sent at a suitable rate.

The SpaceWire IP Tunnel application provides the man-machine interface, in addition to providing the tunnelling functionality. It is written in Java, which means it will work on all platforms for which there are suitable drivers for the SpaceWire devices, currently Windows and Linux.

The application allows configuration of the Tunnel and provides methods to analyse the traffic which is transferred over the Tunnel, including analysis of any higher-level protocols within the data packets. A plug-in concept is supported to allow analysers for higher-level protocols not already supported to be added to the software.

6. Conclusions

The data rate that can be supported by the Tunnel is clearly dependent upon the speed of the Internet connection between the two ends of the Tunnel. Testing has confirmed that a tunnel can provide an almost transparent connection between two SpaceWire devices, with only the data rate indicating that a normal SpaceWire link is not in use. Time-codes and data packets may be interleaved, and this information is maintained across a Tunnel. The link may be disconnected and reconnected at either end of a Tunnel, and the corresponding action will occur at the other end. The operation of the SpaceWire IP Tunnel software and hardware was demonstrated during the DASIA 2005 conference.

The SpaceWire IP Tunnel allows earlier integration of components helping to reduce costs of spacecraft development.

The authors would like to acknowledge the support of ESA for the work done on the SpaceWire IP Tunnel at the University of Dundee.

7. 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, February 2003.

2. Rosello, J., "SpaceWire Web Page", European Space Agency, http://www.estec.esa.nl/tech/spacewire/.

3. C. McClements, S.M. Parkes, and A. Leon, "The SpaceWire CODEC," International SpaceWire Seminar, ESTEC Noordwijk, The Netherlands, November 2003.

4. S.M. Parkes and C. McClements, "SpaceWire Remote Memory Access Protocol", submitted to DASIA 2005.

5. S. Mills and S.M. Parkes, "Virtual Satellite Integration", Proceedings Data Systems in Aerospace (DASIA) conference, Nice, May 2001.

6. STAR-Dundee, "STAR-Dundee Web Site", STAR-Dundee, http://www.star-dundee.com/.