DESIGN OF A WIRELESS LINK FOR SPACEWIRE NETWORKS

Session: SpaceWire Networks and Protocols

Short Paper

Jean R.Paul and Tanya Vladimirova
Surrey Space Centre, Department of Electronic Engineering, University of Surrey, Guildford, UK, GU2 7XH
Email: j.paul@surrrey.ac.uk, t.vladimirova@surrey.ac.uk

ABSTRACT

A large variety of new applications could be implemented in future space missions by combining the SpaceWire and IEEE 802.11 communication protocols. Converting SpaceWire packets into IEEE802.11 packets presents challenges due to bridging a protocol designed for point-to-point links with a protocol usually operating in an ad-hoc mode.

This paper presents an approach to developing a wireless interface for SpaceWire on-board networks for the purpose of inter-satellite communication in networked distributed satellite systems. The tradeoffs of a bridge design supporting SpaceWire/IEEE802.11 data transfers are discussed.

1. INTRODUCTION

Future networked distributed satellite systems (DSS) will be formed of satellites with cross links and on-board autonomy capabilities. In these systems, a spacecraft can gather information from and aggregate its resources with other spacecraft to perform tasks. In such a context, inter-satellite communication is an important feature of the satellite network [1, 2]. A distributed satellite system for space weather monitoring is proposed in [3], in which satellites are able to exchange data via an inter-satellite link (ISL) and a master node is selected to communicate with ground. In case if the master satellite fails, the network will need to reconfigure in order to select another node as master or incorporate a new member satellite.

SpaceWire is a recently developed on-board spacecraft communication protocol, which has already been deployed in a number of space applications. In contrast, the IEEE802.11 wireless standard is a mature terrestrial protocol that offers a wide range of services. Both standards have attributes supporting scheduling, flow control and buffering that can be exploited to provide a communication medium enabling high-speed fault-tolerant wireless networks. SpaceWire has a good electromagnetic compatibility (EMC) performance, is easy to implement, and supports fault-tolerance by providing redundancy to networks. The use of SpaceWire as the communication protocol for intra-spacecraft components can improve robustness and reconfigurability by adding fault-tolerance on board each spacecraft within a DSS network. These attributes make SpaceWire an attractive communication protocol on board spacecraft in distributed satellite systems [3].

This paper presents an approach to developing a wireless interface for SpaceWire on-board networks for the purpose of inter-satellite communication in future networked DSS [2, 3]. The work presented in this paper is concerned with the design of a bridge to connect a SpaceWire router to a Wireless transceiver. The bridge contains a module that is used to convert SpaceWire format into IEEE802.11 packets. A synchronization module is included to manage the flow between the router and the bridge whilst the Gaisler GRLIB memory controller [4] provides off-chip access to the SpaceWire router and modules.

2. SYSTEM OVERVIEW

A high-performance computing reconfigurable system-on-a-chip (SoC) design is proposed in [3] to support data processing and inter-satellite communication on board satellites. The SoC is centred around the LEON3 processor, which is used to run software applications and AMBA2 is used as the on-chip bus system as shown in Figure 1. An IEEE802.11 transceiver intellectual property (IP) core is also included for inter-satellite communication. As IEEE802.11 based communications are asynchronous, a direct memory access (DMA) core is added to the design to control the data transfers between the memory and the wireless transceiver.

SpaceWire is a high-bandwidth and fast switching protocol in which link connection, as well as error recovery, takes 20 microseconds. The protocol is flexible in terms of network topology, there is no limitations in the packet size, and the data rate is only constrained to receiver's buffer size. In contrast, the IEEE 802.11 standard is a contention-based network protocol and, depending on the standard, the bandwidth is limited to 20 or 40 MHz. Packet size is limited to 2310 bytes and the link setup between two links in IEEE 802.11 nodes takes a minimum of 250 µs to start. In that time a SpaceWire node could send in excess of 200 data characters, when operating at 20 MHz. As a result, buffer management and storage are required to avoid bottlenecks in the SpaceWire networks while the wireless transceiver starts up a link. The DMA is able to connect AMBA ready devices to the memory controller, and the bridge interfaces with the DMA via the AMBA AHB bus.

3. BRIDGE DESIGN

3.1 Link Connection and Synchronisation

The IEEE 802.11 wireless network standard, often referred to as WiFi, is defined at the Medium Access Control (MAC) and Physical (PHY) layers [5]. The IEEE802.11 physical layer can either be based on the Orthogonal Frequency Division Multiplexing (OFDM) or spread spectrum. Due to bandwidth scarcity in wireless networks, a common approach is to use a multiple access scheme to share the bandwidth of a communication link between several nodes. The MAC layer ensures that frames are delivered error free, and adds addressing information to the transmitted frames.

Inter-frame timing constraints are introduced at the IEEE 802.11 MAC layer in order to support high data rates. Before a node is allowed to initiate a transmission, it senses the channel to verify whether it is free for a predefined minimum period called Distributed Inter Frame Space (DIFS). If the channel is busy, a random back-off interval is calculated to determine the waiting time before the sending node tries to access the channel again. This is followed by a flow control mechanism between the sending and receiving nodes. SpaceWire has a flow control for link connection as well, however once the link between two nodes is established, the data transfer rate depends only on the receiving buffer size. Typically, a SpaceWire node has a buffer able of storing 7 characters. Before a data transfer, the sender checks whether the receiver's buffer is full and data transfer will not occur until the receiver sends an authorisation to transmit to the source node.

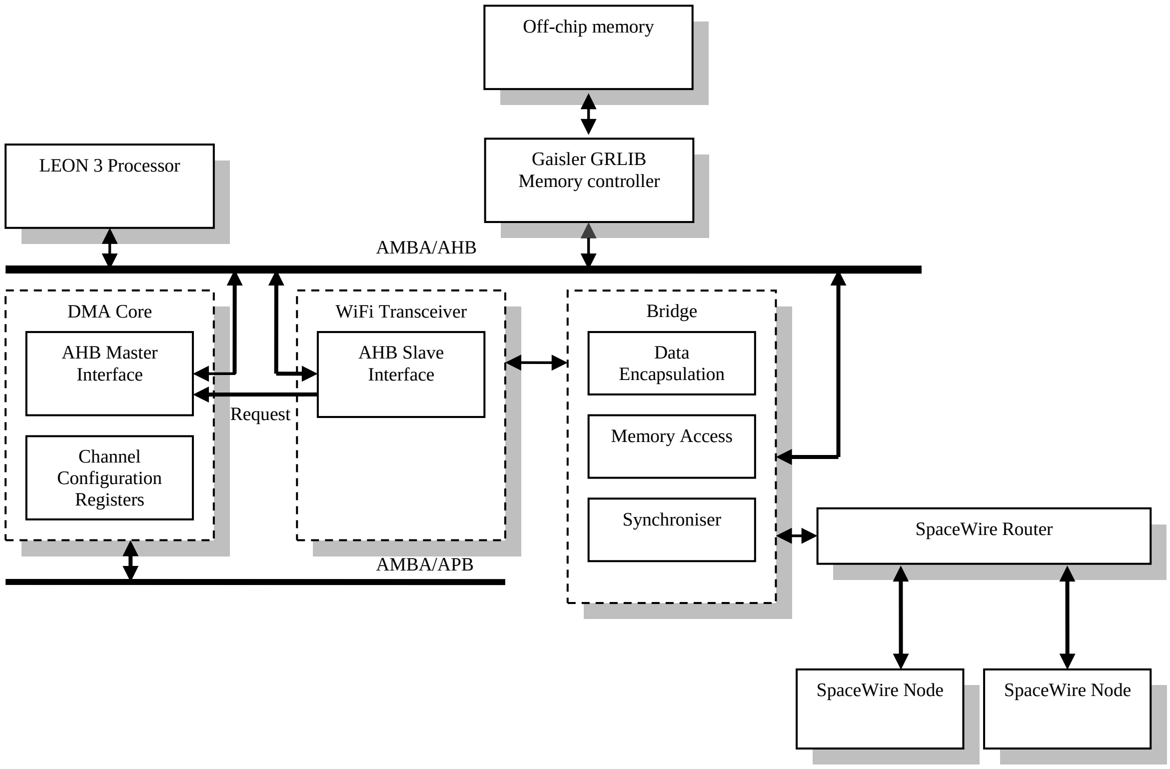

Figure 1: Integration of SpaceWire with the IEEE802.11 standard in a SoC design

In the proposed design, the bridge must keep the SpaceWire router synchronised with the WiFi transceiver. Since the data rate of the OFDM-based IEEE 802.11 nodes is set at 6 Mbps and the bandwidth is 20 MHz, for ease of implementation, the SpaceWire router is also limited to 20 MHz. The IEEE 802.11 wireless transceiver in Figure 1 is implemented as a hardware accelerator in VHDL supporting the highest data rate. Its MAC layer contains functions such as 'byte by byte' processing in both receive and transmit directions, cyclic redundancy check (CRC) generation for error detection purposes, signals to indicate successful transmissions and reception. It was observed that when the data rate is set to 6 Mbps, if a frame is received by the transceiver, the MAC layer forwards the bytes to the bridge at an interval of 350 ns and the SpaceWire destination router's buffer is never full. Even at higher data rates, the MAC would be forwarding data at the same interval which suggests that the bridge is able to efficiently transfer data between the router and a wireless transceiver operating at the highest speed for the IEEE 802.11a, g, and n standards.

3.2 Data Encapsulation

Data coming from the SpaceWire router are presented to the bridge in groups of 9 bits, in which the MSB represents the control bit, a '0' is used for data characters and a '1' is used for end of packet (EOP) and error end of packet (EEP). In order to ensure SpaceWire packets size are within the maximum 2312 bytes packet size supported by IEEE 802.11, a packet size is set in the SpaceWire network. As the wireless transceiver's MAC layer processes on a byte-by-byte to perform CRC, the SpaceWire character's control bit is stripped off in order to comply with the 8-bits input of the MAC layer. The bridge placed at the receiver re-inserts the control bit and the end of packet marker. And when the MAC layer detects an error in the packet by CRC, the MAC layer informs the bridge, which in turns sends error end of packets (EEP) characters to the SpaceWire router.

3.3. Remote Memory Access

The latency involved in establishing a link between WiFi nodes and the use of the transceiver by software applications can lead to a bottleneck in the SpaceWire network. Therefore a mechanism for data storage while the router is waiting for its turn is required. The Remote Memory Access Protocol (RMAP) is proposed to allow SpaceWire nodes to access memory directly [6]. This protocol specifies acknowledged/non-acknowledged and verified/non-verified implementation methods for read and write operations. RMAP is an application layer protocol, in which CRC is performed at both the transmitter and receiver to ensure that data is written error-free in the memory.

Functions such as CRC checking and memory access via the DMA provide the WiFi transceiver with features that present similarities to RMAP. In IEEE802.11 networks an Acknowledgement (ACK) packet is sent to the transmitting node when the destination node receives an error-free. If an ACK is not received within a duration specified by the transceiver's acknowledgement time, the transmitting node will assume that an error or collision occurred.

The long propagation delay in space is a major cause for decreased throughput. This would be further exacerbated by having to exchange a packet between spacecraft to ensure the receiving router is running and, once the SpaceWire is deemed connected, data transfer can be initiated. This may not be an advantageous flow control process, as the latency involved in WiFi link connections would be in the order of hundreds time higher in comparison to the latency of SpaceWire networks. Also, a transmitting SpaceWire node can send erroneous data in case of transient or intermittent faults [7], which would result in an inefficient use of the ISL. In the transmission of SpaceWire packets via the ISL, a link is instead assumed to be connected when a router is ready to send data to another router via the IEEE 802.11 wireless transceiver. This approach presents some distinctive advantages where even if the data originated from the SpaceWire network is error-free, they are still susceptible to the effect of the wireless communications channel effects. The transmission of a WiFi packet of 1500 bytes has a duration of 2 milliseconds and if during that period the router is still disconnected, the link can be considered to be down. The CRC performed by the WiFi MAC layer is able to detect failures due to channel impairment as well as errors in SpaceWire packets. Thus when the source node receives an Ack, channel effects are mitigated and the SpaceWire packet is error-free.

The bridge transfers data from SpaceWire nodes to the off-chip memory via the DMA. When an error is detected, the MAC layer informs the bridge, which in turns appends an EEP to the SpaceWire packets. As opposed to RMAP, the CRC checking is done at a lower level than the application layer, this in turn reduces the latency and the complexity involved in implementing direct memory access into memory-mapped SpaceWire nodes. The DMA is designed with the flexibility of allowing the off-chip memory access to be either hard-coded or set in a C program.

In essence, the bridge is capable of spoofing the end to end SpaceWire link connections. The resulting drawback is a drop in throughput in instances where a router is not connected.

4. CONCLUSIONS

This paper outlines a novel bridge design to connect SpaceWire routers with an IEEE802.11 transceiver. The similarities between the two standards are exploited and the design of a bridge capable of converting from the one standard format to the other is presented. To the best of the authors' knowledge a bridge design permitting the translation of SpaceWire packets into WiFi packets has not been previously proposed. The bridge is also able to control information flow and has a mechanism to provide memory access to SpaceWire routers. The bridge is incorporated into a high-performance SoC, which provides a communication platform enabling spacecraft with SpaceWire networks to communicate via inter-satellite links based on the IEEE 802.11 wireless network standard.

5. REFERENCES

1. K. Sidibeh and T. Vladimirova. "A Fault Tolerant High Speed Network for Inter Satellite Links", Proceedings of 8th Military and Aerospace Applications of Programmable Logic Devices and Technologies International Conference (MAPLD'2005), Washington DC, US, NASA ,September 7-9, 2005, P-144,.

2. T.Vladimirova and K.Sidibeh. "WLAN for Earth Observation Satellite Formations in LEO", Proceedings of 2008 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS'08), Edinburgh, IEEE Computer Society Press, , August 4-6 2008,pp. 119-124.

3. T. Vladimirova, C. P. Bridges, J. R. Paul, S. A. Malik and M.N.Sweeting. "Space-Based Wireless Sensor Networks: Design Issues", Proceedings of 2010 IEEE Aerospace Conference, Big Sky, MT, 6-13 March 2010.

4. GRLIB IP Library User's Manual, http://gaisler.com/products/grlib/grlib.pdf(last accessed 01/04/2010)

5. IEEE 802.11 Wireless Local Area Networks, http://ieee802.org/11/

6. S.M. Parkes, S. Mills, and C. McClements,"SpaceWire Higher Layer Protocols," Proceedings of International Astronautical Congress, Valencia, Sept 2006.

7. S. Parkes, "SpaceWire for Adaptive Systems", Proceedings of NASA/ESA Conference on Adaptive Hardware and Systems, 2008, Page (s) 77-82.