Guarantee of Information Delivery in a Spacecraft Onboard Network Based on the SpaceWire Interface

A.Yu. Koshelev, *koshelev.ay@spacecorp.ru*
Joint Stock Company "Russian Space Systems", Moscow, Russian Federation
A.M. Dianova, *dianova.am@spacecorp.ru*
Joint Stock Company "Russian Space Systems", Moscow, Russian Federation
D.O. Petukhov, *petukhov.do@spacecorp.ru*
Joint Stock Company "Russian Space Systems", Moscow, Russian Federation

Abstract. Nowadays SpaceWire is regarded as a prospective aerospace data transmission interface standard. This paper gives an analysis of the guaranteed information delivery of SpaceWire networks. The result of the analysis showed key problems that did not allow using this standard in real-time networks. A method was developed and proposed to guarantee information delivery in the SpaceWire network based on scheduled routing. Tests using a software model of the modified network were performed. The results obtained confirmed the efficiency of the method used to ensure the guarantee of information delivery in the network. It is shown that a network with a modified protocol stack meets real-time requirements.

Keywords: SpaceWire, wormhole routing, real-time network, scheduled routing, information delivery guarantee

Introduction

The requirements for modern data transmission interfaces in onboard space systems largely correspond to real-time system (RTS) requirements. That is, a temporal determinism requirement is applied [1], meaning the need for the determinism of the interaction time between network elements. In earlier generations of spacecraft onboard equipment (SC OE), RTS requirements were satisfied by the discipline of the computing process in separate nodes and components of complexes and SC OE systems. Yet, with the growth of the functional and structural complexity of SC OE and SC as a whole, the need emerged to implement an advanced OE architecture with high-speed data transmission interfaces that call for the use of unified approaches to meeting RTS requirements in the onboard network. During the development of promising SC OE, the SpaceWire ECSS-E-ST-50-12C (SpW) [2] interface is actively implemented. SpW is a high-speed communication technology similar to the functionality of the Ethernet protocol. Figure 1 shows a comparison of the OSI protocol stack [3], SpW and the protocol stack of a TCP/IP and Ethernet-based in-home network.

The levels of the protocol stack are implemented in the SpW standard up to the network level but there are options for supplementing the SpW protocol stack with a transport level in additional highly specialized standards. However, they are either not suitable as a network-wide protocol for the SC OE control network or require the modification of SpW terminal devices (TDs). Messages are transmitted to the SpW via routes from the source to the receiver of the message. The packet size has no limitations. Logical, path, and regional logical addressing are provided for.

Although the SpW protocol stack is initially intended for aerospace application, it has some inherent drawbacks that lead to problems when this interface is widely used in loaded onboard networks. Hence, attempts are made to propose ways of augmenting the SpW protocol stack for meeting RTS requirements in modern SC OE. Since the present paper is concerned with the augmentation of the SpW protocol stack, the problems of the physical level of the protocol stack are factored out. A number of SpW problems are discussed in the professional community, which are not considered in the standard, but are fundamental in the development of complex systems and SC OE systems based on it. For example, [4] considers a number of problems of the SpW interface. Among

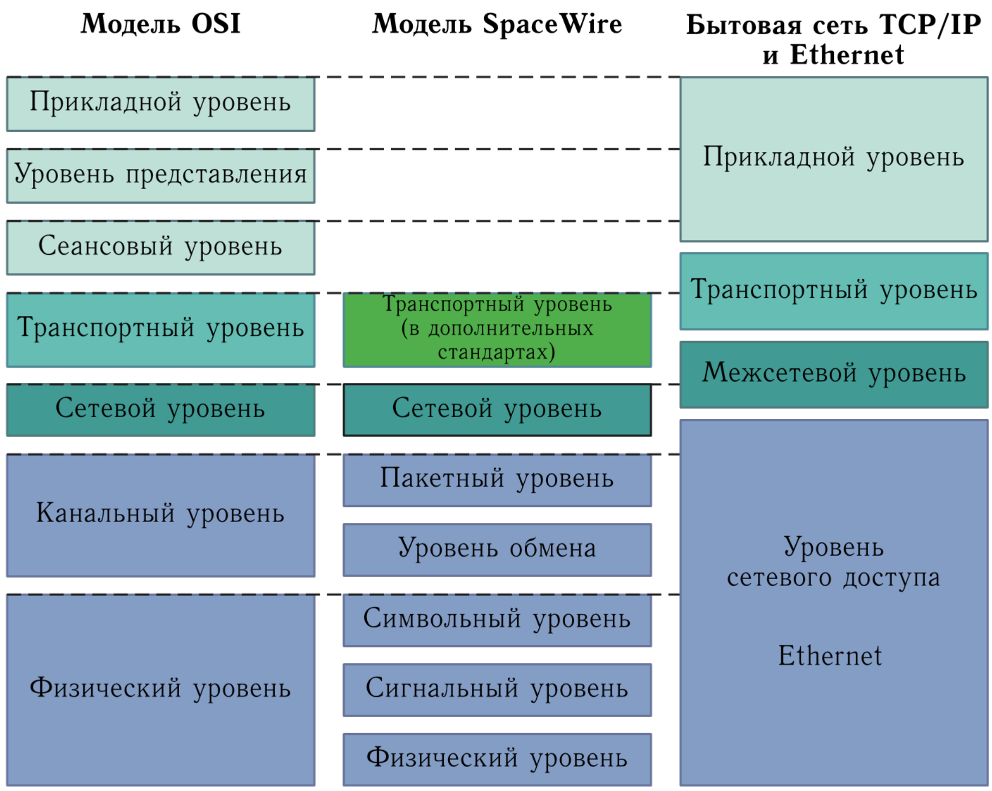

Fig. 1. Correspondence between hierarchical levels of OSI, SpW and TCP/IP models.

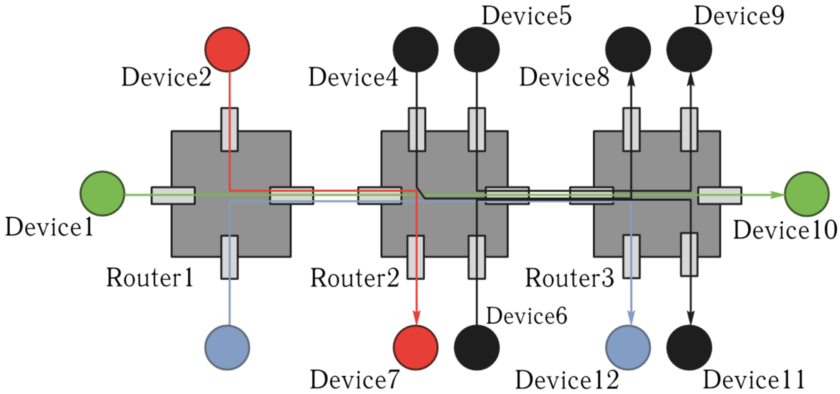

Fig. 2. Topology and data transfer scheme used for simulation.

Table 1. Parameters of transmitted messages

Message transmission number	Source	Receiver	Message size	Minimum time of transmission through the link, ms	Message transmission period, ms
1	Device1	Device10	1 kB	1	8
2	Device2	Device7	64 bytes	0.064	10
3	Device3	Device12	64 bytes	0.064	12
4	Device4	Device8	1 kB	1	32
5	Device5	Device9	1 kB	1	16
6	Device6	Device11	1 kB	1	32

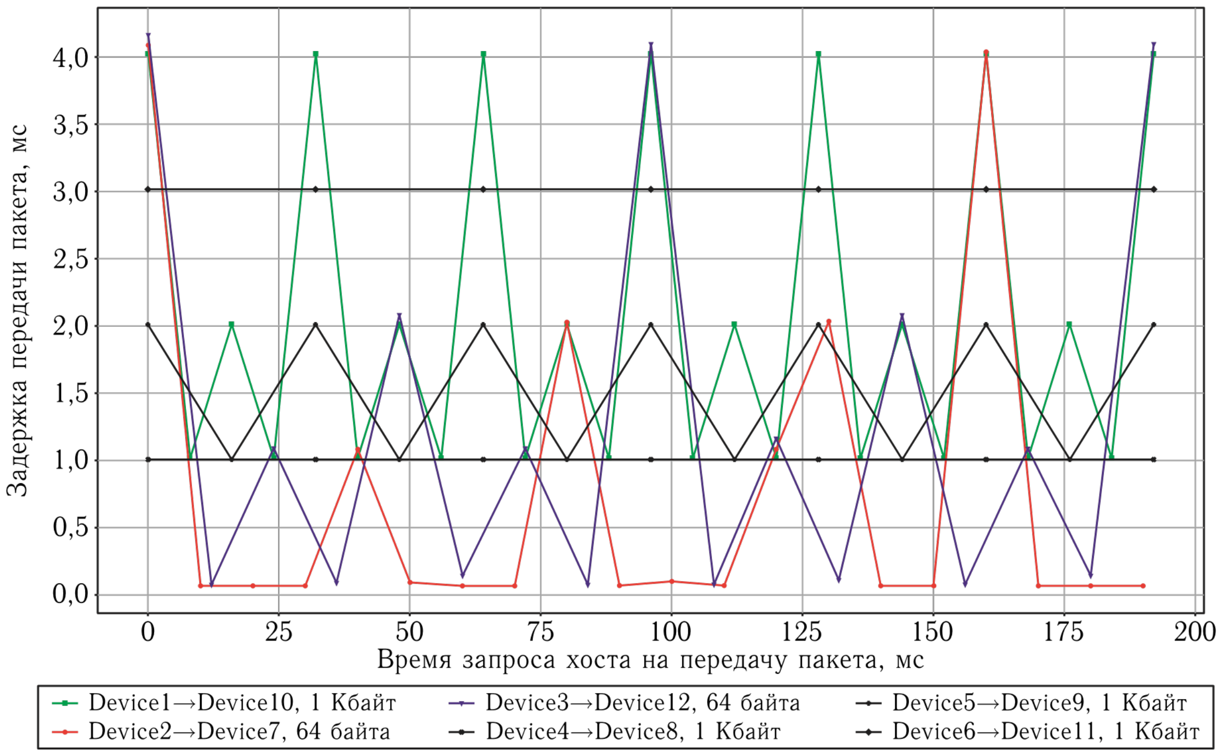

Fig. 3. Delays occurring during network operation.

key issues, critically affecting the quality of network operation is wormhole routing (or wormhole switching). When considering the interface in accordance with the OSI model, this type of routing is implemented at the network layer of the protocol stack [5]. Historically, such a routing method was implemented in multicomputer networks and originates from wormhole switching [6] that is employed for building multiprocessor systems, such as transputers. The implementation of wormhole routing in the SpW protocol stack inherited the problems of wormhole switching. When a packet header is successfully transmitted to the network, the remaining data is sent immediately and the data transmission channel remains blocked until the entire packet is transmitted. Thus, intermediate buffering in the routing switches (RSs) is not required, which significantly reduces the memory requirements of switching devices in comparison to other routing methods. However, the simplicity of the wormhole routing principle introduces a significant drawback – propensity for mutual blocking: the transmission of a large packet can lead to long transmission delays for other packets in the network. Apart from this, an error in the header and physical node failures can lead to blocking. Systematically occurring blockings make it impossible to guarantee the delivery of packets and the time of packet delivery. This does not correspond to the principles of RTS, which are important for the implementation of responsible application systems.

Simulation of a standard SpW network

Network collisions in the SpW network were reproduced using a program-mathematical model of a SpW network, made in accordance with the standard described in [2]. Modeling involved logical protocols of the exchange, packet and network levels, as well as of the simplified character level, regulated by the standard, without going into the physical component. The software simulator is a behavioral model of a system based on the SpW standard given in [2]. The model allows us to simulate data transfer between devices connected via duplex channels (links). The tasks to be solved include:

- simulation of initialization and establishment of a communication link between devices prior to data exchange;
- simulation of device behavior based on the logical protocols of the exchange, packet and network levels, as well as the simplified character level regulated by the standard without delving into the physical component;
- simulation of data exchange mode between terminal devices connected to each other with routing switches.

With the help of the software, model tests were conducted with the imitation of data transmission in a SpW network with configurations leading to packet blocking. The topology with three routing switches (RSs) and twelve terminal devices (TDs) used for testing is given in Fig. 2.

Simulation was conducted for an exchange rate in the entire network of 10 Mbit/s, which does not affect routing principles. The RSs support logical addressing. That is, every RS has its own pre-specified routing table, in which each physical address of the TD is mapped to an output port of the RS.

In view of the determinism of software model functioning, the following scheme of packet transmission order was implemented in the PC: within one RS, the first packet to be transmitted from the TD is the one with a lower sequence number (physical address), and then packets from the ports connected to ports of other RSs are transmitted. Table 1 shows the parameters of transmitted messages.

Figure 3 shows the timing diagram, the points of which illustrate the point in time of the request of the host to transmit a packet (or the packet generation time) and the delay in the transmission of this packet for six transmissions in accordance with Table 1, obtained as a result of simulation. The host request time corresponds to the specified message-sending periods.

For a packet that does not experience blocking during its transmission over the network, the delay time is close to the minimum transmission time through the link. For such packets, all points lie on a single line with the ordinate corresponding to the minimum transmission time through the link. Packet 4 (black in the graph) demonstrates this. The transmission delay of packet 6 is stable but exceeds the minimum time by three times due to the constant blocking of transmission by other packets. The change in delay of packet 5 is periodic. The transmission period of this packet is two times less than that of packet 4, which is ahead in the processing queue. If simultaneous host requests are made, packet 5 is blocked, waiting for packet 4 to finish transmission. Packet 1 (green) passes through two RSs and is periodically blocked by one or two black packets.

Short packets 2 (red) and 3 (blue), which imitate short commands from the control network, are of particular interest. Packet 2 passes only through one RS, however, regular blockings exceeding the minimum transmission time by more than 60 times occur because of the need to wait for the completion of the transmission of the black and green packets. Packet 3 passes through two RSs and is blocked because it has to wait for the completed transmission of black packets, as well as the red and green ones.

The results confirm the outlined disadvantages of wormhole routing that lead to inconsistent delays in information transmission, which complicates the implementation of SpW in system or control loops for critical applications that are subject to real-time system requirements. The results of software simulation are validated by simulation on real hardware. Table 2 shows the test results. In this network, packets of 4 kilobytes were transmitted to one port and a response acknowledgment receipt was received at a link speed of 10 Mbit/s.

Table 2. Hardware test results

Source cycle *S*, ms	Acknowledgement from source 1 wait time, ms	Acknowledgement from source 2 wait time, ms
20/20	2.5–10	10–40
20/0	2.5–8	10–30
0/0	2.5–1000	10–50

The first column of the table shows the periods of message transmission. The range of delays of the acknowledgement wait time (small packet) is conditioned by the asynchronous operation of the TDs and the operation logic of the DMA (Direct Memory Address) controller, which is similar to the conditions of OE application. The performed tests confirmed the existence of problems concerning the use of SpW as a control network.

The indicated disadvantages of the SpW protocol stack can be overcome by the application of several methods, namely:

1) imposing additional requirements on the discipline of exchange of each network terminal device during SC OE development;

2) limiting the information load on the network to a level of 10–15%; and

3) augmenting the SpW protocol stack, which implements a different routing principle in the network switch, but at the same time one that does not require the modernization of terminal devices complying with the SpW standard [2].

The present paper focuses on the option of augmenting the standard SpW protocol stack. In order to validate the proposed concept, software and mathematical simulation was conducted both in accordance with the existing SpW standard and with a modified switch.

Simulation of a modified SpW network

As mentioned above, a transport level can be added to the SpW network within the framework of the protocol stack concept of the OSI hierarchical model. The transport level should introduce coherence of RS operation into the SpW network, which allows observing the conditions for the transmission of real-time network data. The proposed principle is a synchronous SpW network with fixed-size data (frames) transferred at certain time intervals (time slots) according to a time schedule, in which each transmitted frame (packet) corresponds to a time slot on every port of every RS. The schedule, routes and delays for frame transmission are calculated in advance, thus eliminating the occurrence of collisions in the network. Apart from this, a transmission schedule allows the use of the network resource with maximum efficiency and the correct estimated multicast delivery similar to Time-Code transmission in accordance with SpW.

The proposed logic of the transport level protocol is simple enough and does not require serious computing resources. The task of the transport level protocol is to make sure the message is sent strictly according to the specified schedule, which ensures the behavioral and temporal determinism of each RS in the network. For this, each RS has its own “message exchange schedule” that regulates all exchange operations that pass through the RS ports. Thus, at each given time interval, a transmission of only one specific message takes place through each port of the RS.

The operation of a modified SpW network was simulated on the developed scalable software model of the network. This model employs a similar simulation approach without delving into the character and physical levels of the exchange. The same test case was used to simulate the network as in Fig. 4.

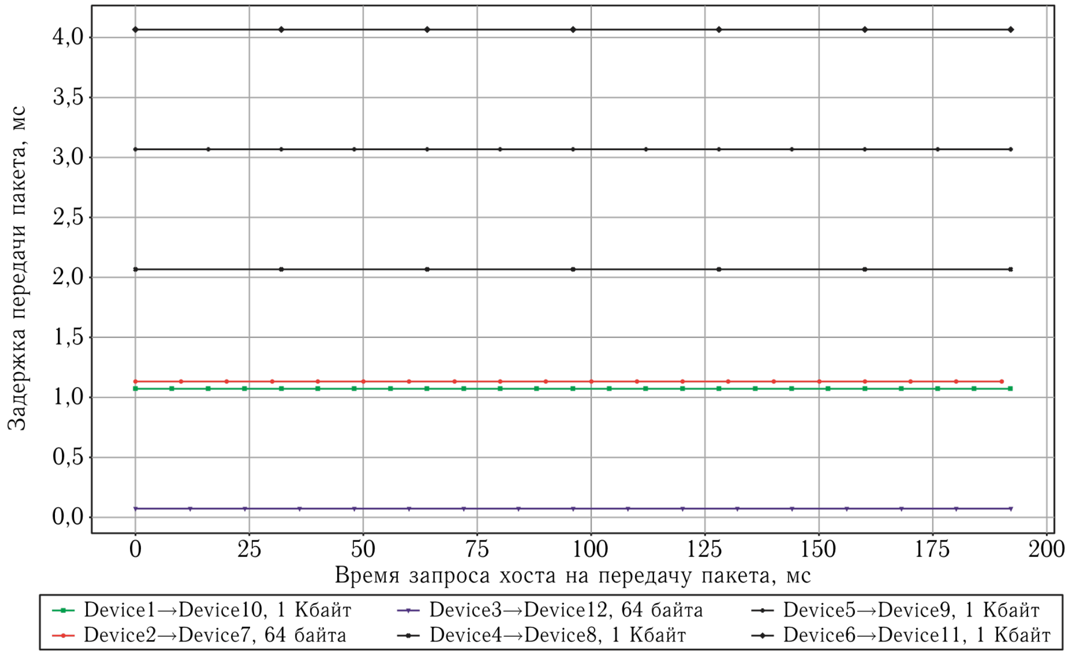

Fig. 4. Results of the operation of the modified network.

The results of simulating a modified network are shown in Fig. 4. Packet delays in this case are predictable, constant and are within acceptable limits. Thus, the network meets RTS requirements.

Conclusion

Scheduled routing can help control blockings in the onboard network, as well as maintain system determinism and provide guarantees of message delivery with fixed delays, which is extremely important when implementing control loops. The proposed transport-level routing protocol does not require additional logic from SpW terminal devices and, therefore, is capable of functioning in the networks of existing devices. Another advantage of such an approach to network organization is the possibility of hot or cold redundancy of channels by setting different routes (virtual channels) when creating a schedule or a set of schedules for different system operation modes. In return, such a scheme requires a mandatory source of time messages (Time-Code) together with the implementation of a buffer memory for each RS port. As for the development of this transport-level protocol, it is possible to propose the allocation of part of the network bandwidth for asynchronous information transmission with wormhole routing over the network that is insensitive to delays and collision-induced data loss.

References

1. Burdonov I.B., Kosachev A.S., Ponomarenko V.N. *Operatsionnyye sistemy real'nogo vremeni* [Real time operating systems]. Moscow, Institut sistemnogo programmirovaniya RAN, 2006, 49 p. (in Russian)
2. *ECSS-E-ST-50-12C, SpaceWire – Links, nodes, routers and networks*. ECSS Secretariat ESA-ESTEC, Requirements & Standards Division, Noordwijk, The Netherlands, 31 July 2008. European Cooperation for Space Standartization. Available at: https://ecss.nl/standard/ecss-e-st-50-12c-spacewire-links-nodes-routers-and-networks/ (accessed 08 October 2020).
3. *ISO/IEC 7498-1:1994, Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*. ISO/IEC JTC1, June 1996, 59 p. International Organization for Standardization. Publicly Available Standards. Available at: https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (accessed 08 October 2020)
4. Zhuravlev V., Nemytov A., Osipov Yu., Pershin A. SpaceWire: vzglyad so storony. Chast' 1 [SpaceWire: A look from the outside. Part 1]. *Sovremennaya elektronika: zhurnal dlya spetsialistov* [Modern electronics: Magazine for specialists], 2017, No. 8, pp. 36–40. Available at: https://www.soel.ru/magazines/PRINT/SoEl_2017-8/SoEl2017-8pr.html (accessed 08 October 2020). (in Russian)

5. *Proyekt Rossiyskogo Standarta SpaceWire-Rus, sovmestimogo s mezhdunarodnym standartom SpaceWire (SpaceWire-ECSS-E-ST-50-12C)* [Project of Russian Standard SpaceWire-Rus compatible with the international standard SpaceWire]. SPBGUAP, 2012. (in Russian).
6. Mohapatra P. Wormhole routing techniques for directly connected multicomputer systems. *ACM Computing Surveys*, Vol. 30, No. 3, September 1998. Available at: https://www.cs.hmc.edu/~avani/wormhole/1998-p374-mohapatra.pdf (accessed 08 October 2020).