STP-ISS Transport Protocol Overview and Modeling

Yuriy Sheynin, Valentin Olenev,
Irina Lavrovskaya, Ilya Korobkov
Saint-Petersburg State University of Aerospace Instrumentation
Saint Petersburg, Russia
sheynin@aanet.ru, {valentin.olenev, irina.lavrovskaya, ilya.korobkov}@guap.ru

Sergey Kochura, Sergey Openko, Dmitry Dymov
JSC "Academician M.F. Reshetnev" Information Satellite Systems"
Zheleznogorsk, Russia
{kochura, openkosi}@iss-reshetnev.ru, dymovdv@mail.ru

Abstract—**The paper is concerned with the development of the Transport protocol "STP-ISS" for SpaceWire networks. Firstly we give a brief overview of transport protocols which are intended to operate over SpaceWire: RMAP, CCSDS PTP, STUP, JRDDP and STP. Hereinafter, the paper presents the first revision of the new STP-ISS transport protocol which was developed in accordance with the results of the overview and technical requirements to the transport protocol. Finally, we describe STP-ISS modeling, which was very efficient for the STP-ISS protocol development, testing, analysis and improvement. This paper describes three modeling directions: C++ reference code, SDL-specification and SystemC network model.**

I. Introduction

SpaceWire is a data-handling network for the spacecraft, which combines simple, low-cost implementation with high performance and architectural flexibility [1]. MIL-STD 1553 has been the communications standard for spacecraft and avionics for a long time. 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. So new technologies are being actively integrated into new spacecrafts, and SpaceWire is one of them. SpaceWire is now being used on more than 30 high profile missions and by all of the major space agencies and space industry over the world.

The basic SpaceWire protocol standard covers three bottom layers of the OSI model and does not provide transport services. The most evident solution is using TCP/IP [2] as it is an open-standard protocol which is widely used in terrestrial networks. Although the use of TCP/IP provides many benefits, the standard implementation may not be suitable for use with SpaceWire onboard spacecraft because of the high overhead it imposes on small packets [3].

There are a number of transport protocols that had been specifically developed to operate over SpaceWire. The detailed overview and analysis of them were presented in our previous article on STP-ISS protocol development, which was firstly provided at FRUCT-15 by the paper "Analysis of the Transport Protocol Requirements for the SpaceWire On-board Networks of Spacecrafts". In the abovementioned paper we gave a detailed overview and comparison of RMAP, CCSDS PTP, STUP, JRDDP and STP protocols [4].

TABLE I. Protocols comparison

Protocol / **Feature**	**RMAP**	**PTP**	**STUP**	**JRDDP**	**STP**
Configuration flexibility	√	–	–	–	√
Multiple applications	–	–	–	√	√
Data flows of different priorities	–	–	–	√	–
Data flow control	–	–	–	√	√
Transport connection establishment	–	–	–	√	√
Segmentation	–	–	–	√	–
Data correctness check	√	–	√	√	√
Data sequence check	–	–	–	√	–
Data retransmission	–	–	–	√	–
Acknowledgements	√	–	–	√	–
Scheduling	–	–	–	–	–

General features of these protocols are given in Table I. According to the conducted analysis, nowadays, there is no transport protocol operating over SpaceWire which could provide different types of quality of service, guaranteed data delivery and configuration flexibility. Currently existing protocols which were designed specifically for SpaceWire are not dedicated for the stated tasks. RMAP provides rich means for switch configuration and monitoring of network status [5]. The CCSDS PTP protocol is primarily intended for space packets encapsulation into the SpaceWire packets for their further transmission over the network [6]. STUP and STP protocols [7], [8], similarly to CCSDS PTP, do not support any quality of service except best effort. In turn, the JRDDP protocol provides reliable data delivery and uses priorities but this protocol does not provide guaranteed services [9]. Also, we should mention the SpaceFibre technology, which was developed in the scope of the FP7 SpaceWire-RT project [10]. SpaceFibre covers only link and physical layers and defines very strong quality of service facilities. The QoS and FDIR layer of SpaceFibre provides quality of service and flow control for a SpaceFibre link. It also provides a retry capability; detecting any frames or control codes that go missing or arrive containing errors and resending them [11]. In our work we took into account some beneficial and useful solutions of SpaceFibre and tried to bring them into transport layer mechanisms.

Taking into account the transport protocol analysis and the technical requirements we proposed the first revision of STP-ISS transport protocol. Therefore, current paper gives an overview of the first revision of the STP-ISS transport protocol and describes its SDL specification, reference-code development and network modeling.

II. STP-ISS TRANSPORT PROTOCOL

Fig. 1 shows an example of the on-board network for a small-sized satellite. Dotted lines show the information flows from sensors to the other parts of the satellite. This is just an example of the applied topology, while STP-ISS also can be used for the much more complex networks. The maximum number of the nodes is regulated by the SpaceWire technology.

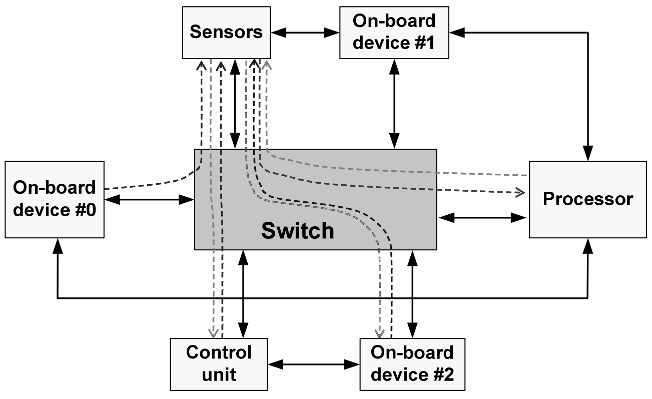

Fig. 1. An example of the onboard network topology

In addition, a network could be divided into several regions. Fig. 2 shows an example of the on-board network with several regions.

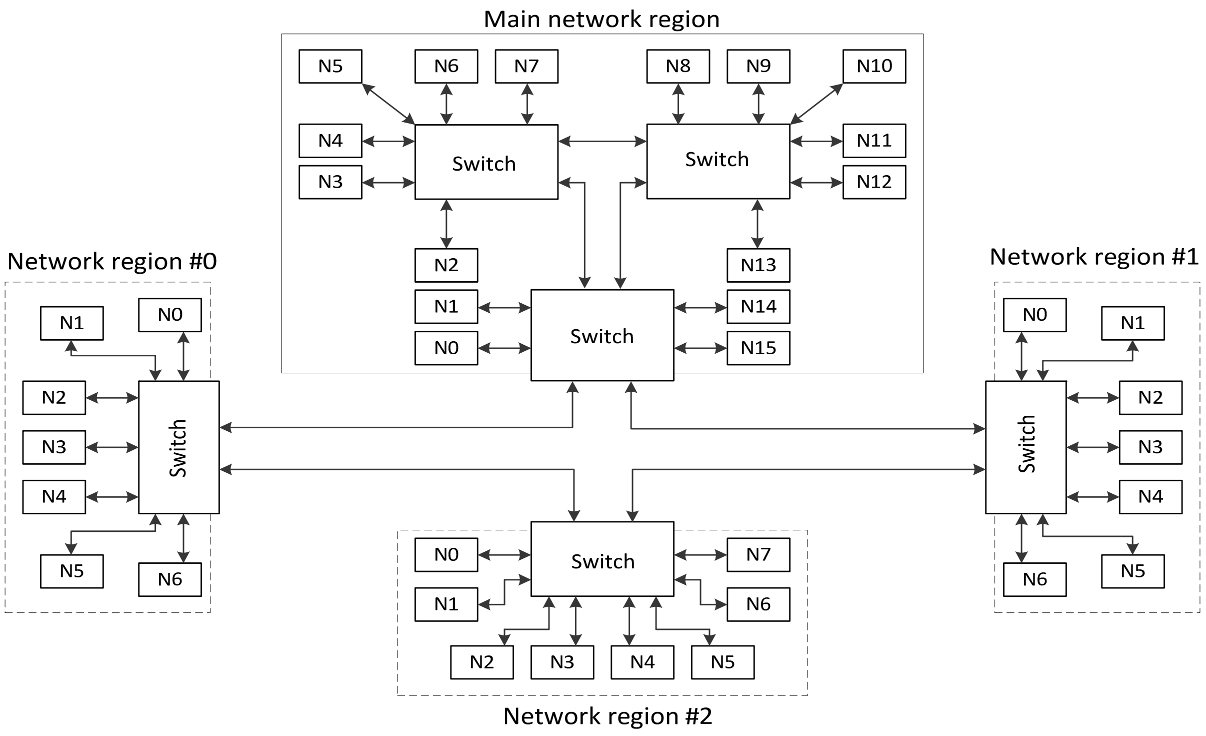

Fig. 2. An example of the onboard network topology with regions

A. STP-ISS general description

STP-ISS is a transport layer protocol that describes informational and logic interaction between onboard devices, packets' formats and packet transmission rules for the SpaceWire network. The onboard software performs functions of Session, Presentation and Application layers according to the OSI model [12]. STP-ISS protocol corresponds to the Transport layer and provides means for transmission of data between the nodes of the network with the required quality of service type and data flow priority. This protocol gives ability for data resending in case of an error detection in the received data. The place of the STP-ISS protocol in the SpaceWire standard's family and conformity to the OSI model is shown in Fig. 3.

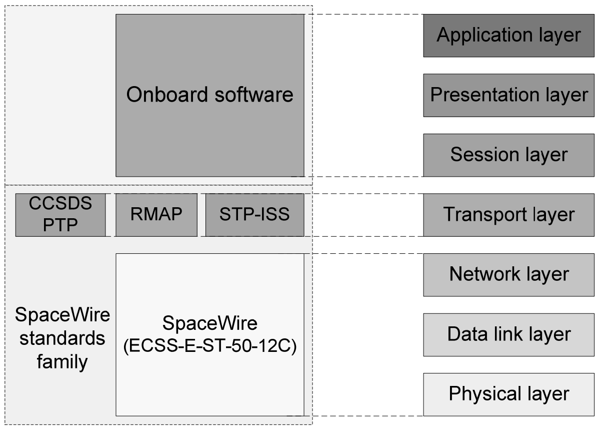

Fig. 3. STP-ISS protocol and OSI model

B. STP-ISS interfaces

There are three interfaces for interaction between the STP-ISS and Applications: Data Interface, Configuration Interface and Control Codes Interface. In addition, there are two interfaces for interconnection with the SpaceWire: SpaceWire packets interface and Control Codes Interface (see Fig. 4).

STP-ISS provides transmission of the following types of data through these interfaces:

- control commands;
- data packets;
- SpaceWire time-codes;
- SpaceWire distributed interrupts and interrupt-acknowledges.

The data interface provides transmission of control commands and data messages. Messages and control commands are transmitted to the remote node by encapsulation into SpaceWire packets.

The configuration interface provides means for the STP-ISS configuration parameters change and for transmission of status information and reset commands.

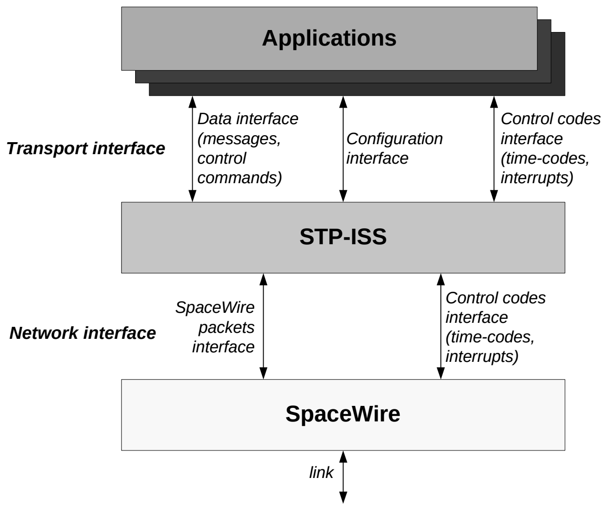
Applications
Transport interface
Data interface (messages, control commands)
Configuration interface
Control codes interface (time-codes, interrupts)
STP-ISS
Network interface
SpaceWire packets interface
Control codes interface (time-codes, interrupts)
SpaceWire
link

Fig. 4. STP-ISS interfaces

The control codes interface passes the SpaceWire time-codes and distributed interrupts to the SpaceWire and then – to the other nodes of the network.

C. STP-ISS application messages

One of the main tasks of the STP-ISS transport protocol is to provide the transmission of messages from the Applications to the remote nodes of the SpaceWire network. The message is a data block that is passed to the STP-ISS from the application layer. There are two types of application messages:

- urgent messages (higher priority);
- common messages (lower priority).

Messages from Applications are encapsulated into SpaceWire packets at the transport layer (see Fig. 5).

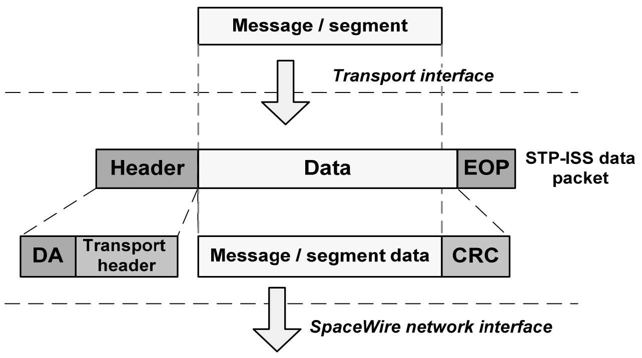

Fig. 5. STP-ISS encapsulation of a message into a SpaceWire packet

The length of each message should not be less than 1 byte and should not exceed 2048 bytes, because the STP-ISS transport protocol does not perform segmentation. Segmentation of messages is done by the application layer and STP-ISS processes these segments as usual independent messages. The Application layer of the remote node assembles the segments into the original message. The message should be assembled basing on the segment identifiers that should be transmitted in the segment header. For this purpose, STP-ISS packet has a secondary header, which should be used by the Application to transmit the information for assembling messages (for example, a number of the segment).

STP-ISS provides the reliable data transmission by using CRC-16 for protection of payload and packet header and for errors detection. CRC-16 covers the packet starting from the first byte of the STP-ISS packet header (excepting path address) till the last byte of data, excluding the end of packet symbol EOP (see Fig. 6).

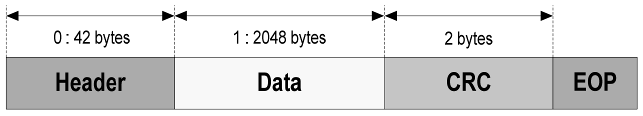

Fig. 6. STP-ISS data packet format

D. STP-ISS lifetime timers

STP-ISS protocol has a special packet lifetime timer, which counts the time, when the packet is still actual in the SpaceWire network. Each packet is stored in the buffer during its lifetime. The value of the lifetime timer is an STP-ISS configuration parameter and it could be set during the configuration stage. Each packet type could have different values of lifetime timer. The lifetime timer should start when the packet is written to the transmitter buffer. The packet should be deleted from the buffer when the lifetime timer expires.

E. Resend buffers

The transmitter side of the protocol has separate buffers for each priority of the transmitted data:

- control commands buffer;
- urgent messages buffer;
- common messages buffer;

The size of these buffers should be set depending on the message or segment size, which the node uses for the data exchange. Also the size of the buffer depends on the type of the device, which implements STP-ISS. The size of each buffer (on the transmitter of receiver side) should not be less then the packet size. However, it is recommended to set the size such a way, that buffer should be able to store two control command packets or two data packets. The resending buffers are shown in Fig. 7.

The data field should not be empty. If the size of data field is 0 then the Application should be indicated that the message is not sent by cause of the zero data length.

The packet should be stored in the buffer until one of the following events occurs:

- the STP-ISS transmitter receives an acknowledgement for this packet using guaranteed quality of service;
- transmission of the packet with the best effort quality of service to the SpaceWire network;
- lifetime timer for this packet expired;
- reset or flush command.

If the buffer overflow occurs, the application should wait until the free space for the message is available.

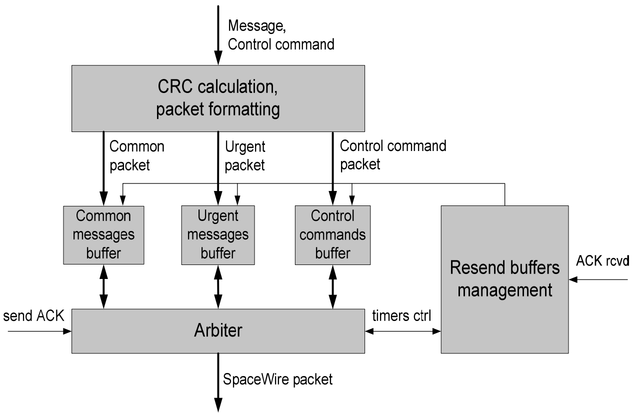

Fig. 7. STP-ISS resend buffers

F. Receiving buffers

The receiver side of the transport protocol has one buffer for all types of the packets, because SpaceWire packets come from the SpaceWire interface sequentially. The size the buffer should be able to store at list two packets.

If the receiving buffer is full, then STP-ISS should indicate the Application layer about it and discard all the packets coming from the SpaceWire.

G. STP-ISS quality of service

One of the benefits of the STP-ISS is the possibility to transmit data using the following quality of service types:

- priority quality of service;
- guaranteed delivery quality of service;
- best effort quality of service.

1) Priority quality of service: Priority quality of service is the main quality of service type that should be supported by all the network end-node devices, which communicate by means of STP-ISS. According to this quality of service type, the data with the higher priority should be transmitted first. Current STP-ISS specification supports 7 levels of priorities:

1) Acknowledgement packets;
2) Control command packets;
3) Resend control command packets;
4) Urgent data packets;
5) Resend urgent data packets;
6) Resend common data packets;
7) Common data packets.

STP-ISS analyses the packet transmission requests during the arbitration. The packet format contains a special flag 'Packet Type' and packet resending attribute. Depending on these, STP-ISS decides which packet should be sent first. The next packet arbitration and transmission starts only after the current packet transmission ends.

2) Guaranteed delivery quality of service: Guaranteed delivery quality of service provides confirmation for the successful packet transmission by sending the acknowledgement packets. Also, it resends the data from the transmitter end-node if the acknowledgement is lost (resending mechanism).

Guaranteed delivery is provided by a number of mechanisms such as resend timers and successful transmission acknowledges. Data resending is based on the packets numeration. This numeration is performed by the application layer by giving an identification number for each packet that is transmitted from a particular application. Therefore, the combination of the application identifier and the packet identification number uniquely identifies each packet.

If a packet is passed to the network layer with the guaranteed delivery quality of service, STP-ISS should start the resend timer for this packet. If a resend timer expires before the receipt of an acknowledgement, this means that the packet or its acknowledgement is lost, or the packet has been corrupted during the transmission. So when the resend timer expires, the corresponding packet should be sent to the network again. Each transmitted packet should have its own resend timer.

The acknowledgement packets are used for confirmation of the packet's successful receipt. The acknowledgement should be sent in the following cases:

- no CRC error,
- the data length field is correct,
- "Guaranteed delivery packet" flag in the received packet's header set to 1.

Within the acknowledgement, the receiver sends a combination of the application identifier and the transmitted packet's identification number. When the transmitter gets the acknowledgement, the corresponding packet should be deleted from one of the transmitter's resend buffers. All the timers associated with this packet should be stopped.

3) Best effort quality of service: Best effort quality of service provides data transmission without sending acknowledges. Such packets have the flag "Guaranteed delivery packet" set to 0 and they do not need resend timers. When STP-ISS receiver gets a best effort packet it checks the CRC and data length, but in case of error or if the packet ends with EEP, data packet still should be sent to the Application, but with an error indication.

H. STP-ISS configuration parameters

The important STP-ISS feature is its configuration flexibility. The protocol has a number of configuration parameters, which give ability to tune the protocol depending on the developer needs (required quality of service, onboard equipment type, resource constrains, etc.). Configuration of the STP-ISS protocol is performed via the configuration interface. Configuration is done in the following cases:

- switching-on/off the device;
- reset;
- switching to the redundant onboard device;
- emergency recovery.

The current STP-ISS specification describes five configuration parameters:

1) Control command lifetime;
2) Urgent message lifetime;
3) Common message lifetime;
4) Resend timeout;
5) Guaranteed / best effort data transmission.

I. Reset and Flush

There are two additional signals that could be passed from the application layer to the STP-ISS through the configuration interface: Reset and Flush. Reset corresponds to the warm reset, and Flush is used for clearing of both transmit and receive buffers.

When STP-ISS gets the Reset command, it should clear transmit and receive buffers, stop all the timers corresponding to deleted packets and set all the configuration parameters to the default settings.

When STP-ISS gets the Flush command, it also should clear transmit and receive buffers and stop all the timers corresponding to deleted packets, but all the configuration parameters should not be changed.

III. STP-ISS MODELING

Once the initial specification of the STP-ISS protocol has been created, it should be precisely analysed, investigated and tested on the early stage of protocol development. In order to check STP-ISS protocol mechanisms we used three different models (see Fig. 8):

- C++ reference code
- SDL model
- SystemC network model

Basing on the results of analysis, testing and simulation, we produced the STP-ISS revision 1 specification.

The reference code is intended to be used as the reference for the programmers, who will implement STP-ISS in the onboard software.

The SDL model is needed for the clear formal description of the STP-ISS internal mechanisms and specification analysis. The SDL specification could be used as a separate document describing the specified mechanisms, and it would be a useful part for the main protocol specification document.

The SystemC model shows the STP-ISS protocol operation over SpaceWire network, and it gives an ability to test the network configuration and test networking features.

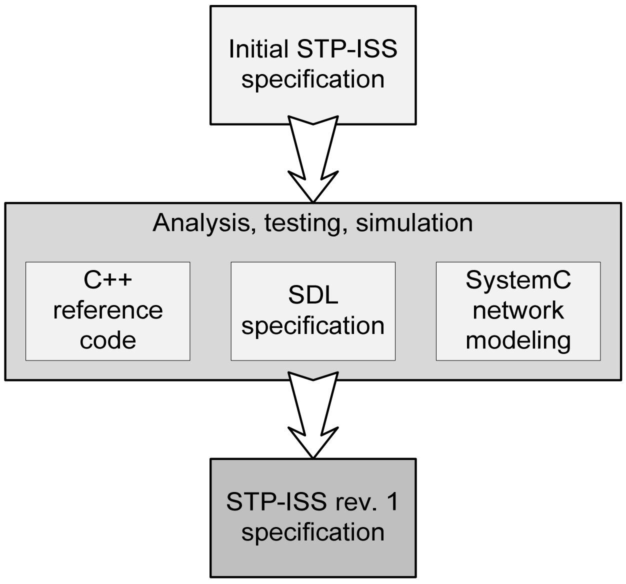
Initial STP-ISS specification
Analysis, testing, simulation
C++ reference code
SDL specification
SystemC network modeling
STP-ISS rev. 1 specification

Fig. 8. STP-ISS specification and modeling

A. STP-ISS reference code

The reference code is a software implementation of the STP-ISS protocol in C++ language. This implementation corresponds to the specification as accurate as it is possible. The C++ reference code describes the logical structure of the protocol, its interfaces and internal mechanisms. All methods which describe protocol functionality are provided with detailed comments for each line. In addition, in order to check and prove the accuracy of STP-ISS the model contains a number of test scenarios for studying and demonstration of protocol functioning. Each scenario launch produces detailed log files with event traces of nodes and of a channel.

This reference code is planned to be used for studying of the protocol functionality. Moreover, it could be translated to the other programming language and used for the implementation of STP-ISS in the onboard software.

B. STP-ISS model in SDL

SDL (Specification and Description Language) is a language for unambiguous specification and description of the telecommunication systems behaviour. The SDL model covers the following five main aspects: structure, communication, behavior, data and inheritance. SDL language is intended for description of structure and operation of the distributed real-time systems. Writing an SDL model of the protocol specification is itself a test of the specification for completeness and unambiguousness. As a result, the consistent readable textual description and formalised specification in SDL are produced [13].

SDL language was used for STP-ISS specification and simulation. The STP-ISS SDL model formally describes all mechanisms, interactions and functionality stated in the specification. We performed verification of the STP-ISS protocol by simulation in IBM Rational SDL Suite. The test system consists of two nodes which communicate via a simple SpaceWire channel. Configuration and generation of test sequences and monitoring was performed by a special Test Engine. This simulation gave an ability to check and verify all internal mechanisms, validate the consistency of the specification and check functional requirements, defined for the standard.

The main advantage of this model is that it is implemented in a formal high-level graphical language. This model can be used for further investigation of STP-ISS protocol.

C. STP-ISS network model in SystemC

SystemC modeling is one of the most efficient and widely used methods for studying, analysis and constructing multi-component systems, such as stacks of protocols, embedded networks of a large number of nodes, systems-on-chip, networks-on-chip, etc. [14] SystemC is a set of C++ classes and macros that provide an event-driven simulation engine. It is specifically designed for modeling parallel systems. This library allows describing multi-component systems and program components, and modeling their operation [15].

The aim of the network STP-ISS model development is to simulate communication of devices (switches and nodes) via the SpaceWire links. The STP-ISS network model consists of the following SystemC modules:

- SpaceWire protocol model, which consists of SpaceWire nodes, switches and channels;
- STP-ISS transport protocol model, which operates over the SpaceWire inside the each node ;
- Traffic Generators, which operate over the STP-ISS models and give ability to launch different tests and generate test sequences.

The SystemC network model provides different configuration parameters for the SpaceWire model such as:

- data transmission speed,
- number of nodes and switches,
- time delay and routing table for the switch,
- number of ports in the switch, etc.

It gives an ability to simulate operation of the various number of devices in a network with different topologies.

D. Simulation results

These three modeling directions were used for the STP-ISS protocol development, testing, analysis and improvement. STP-ISS was checked on conformance to the industry requirements and also on existence of inconsistencies and ambiguities. This work helped to improve a number of algorithms and mechanisms of the protocol, find better solutions and test the STP-ISS and SpaceWire handshake. As a result, we produced a new version of the STP-ISS rev. 1 specification.

CONCLUSION

The paper discusses the development of STP-ISS rev.1 transport protocol for the onboard SpaceWire networks. First of all we give an analysis of currently existing transport protocols intended to operate over SpaceWire. Then the paper gives an overview of main mechanisms of the first revision of the new STP-ISS transport protocol. Finally we discuss three directions under which we investigate, verify and test STP-ISS: SDL specification, reference-code development and network modeling. As a result of the conducted work we produced a new version of the STP-ISS rev. 1 specification.

However, we plan to update the current STP-ISS revision. The following additions are considered to be included to the second revision of the STP-ISS:

1) Scheduled quality of service, when each node of the SpaceWire network will have a permission to send data during the particular time-slot only.

2) Connection-oriented data transmission.

3) Flow control mechanism for each transport connection.

4) Duplicate packets detection.

In addition, we plan, that the second revision of STP-ISS would successfully work with the first revision.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation under the contract №14.578.21.0022.

REFERENCES

[1] ESA (European Space Agency). *Standard ECSS-E-50-12C, "Space engineering. SpaceWire – Links, nodes, routers and networks.* European cooperation for space standardization". Noordwijk: ESA Publications Division ESTEC, 2008. 129 p.

[2] Information Sciences Institute, University of Southern California. RFC 793, Web: http://www.ietf.org/rfc/rfc793.txt. September 1981.

[3] Mills, S. & Parkes, S. "TCP/IP Over SpaceWire", *in Proc. of DASIA 2003* (ESA SP-532). 2-6 June 2003, Prague, Czech Republic. Editor: R.A. Harris. Published on CDROM.

[4] V. Olenev, I. Lavrovskaya, I. Korobkov, D. Dymov, "Analysis of the Transport Protocol Requirements for the SpaceWire On-board Networks of Spacecrafts", *In Proc. of 15th Seminar of Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program,*; Saint-Petersburg: Saint-Petersburg University of Aerospace Instrumentation, 2014. pp. 65-71.

[5] ESA. *Standard ECSS-E-ST-50-52C, SpaceWire — Remote memory access protocol.* Noordwijk : Publications Division ESTEC, February 5, 2010.

[6] *Standard ECSS-S-ST-50-53C, SpaceWire — CCSDS Packet Transfer Protocol.* Noordwijk : Publications Division ESTEC, February 5, 2010.

[7] EADS Astrium GmbH, ASE2. *SMCS-ASTD-PS-001 1.1, STUP SpaceWire Protocol.* July 24, 2009.

[8] Y. Sheynin, E. Suvorova, F. Schutenko, V. Goussev, Streaming Transport Protocols for SpaceWire Networks, *International SpaceWire Conference 2010*, Saint-Petersburg, 2010.

[9] Laboratories, Sandia National. *Joint Architecture System Reliable Data Delivery Protocol (JRDDP).* Albuquerque, New Mexico, May, 2011.

[10] S. Parkes, SpaceWire-RT (SpWRT), *Let's Embrace Space, Volume II*, Luxembourg: Publications Office of the European Union, 2012.

[11] S. Parkes, C. McClements, D. McLaren, A. Monera Martinez, A. Ferrer Florit, A. Gonzalez Villafranca, SpaceFibre Implementation, Test and Validation*, In Proc. Of the 6th International SpaceWire Conferrence*, Space Technology Center, University of Dundee, Dundee, 2014.

[12] Tanenbaum, A. S., *Computer Networks,* Fifth Edition; Prentice Hall, 2011. 962.

[13] International Telecommunication Union, *Recommendation Z100: Specification and Description Language (SDL*), 2007

[14] Open SystemC Initiative (OSCI), *IEEE 1666™-2005 Standard for SystemC*, 2011. 614 p.

[15] D. Black, J. Donovan, B. Bunton, A. Keist, *SystemC: From the Ground Up*, Springer, 2010. 279 p.