Second Revision of the STP-ISS Transport Protocol for On-Board SpaceWire Networks

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, openko}@iss-reshetnev.ru, dymovdv@mail.ru

***Abstract*—The paper provides an overview of the second revision of the Transport protocol STP-ISS, which is developed for SpaceWire on-board networks. Current R&D activity is performed by the specialists of SUAI and JSC “ISS”. The paper shortly compares two revisions of the specification and describes the main mechanisms and quality of service types of the second revision of the protocol. STP-ISS is planned to be used for the next generation spacecrafts.**

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 the leading one. 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 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, but TCP/IP implementation may be not suitable for use with SpaceWire on-board 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 is presented in [4]. In that paper we gave a detailed overview and comparison of RMAP [5], CCSDS PTP [6], STUP [7], STP [8] and JRDDP [9] protocols. According to the transport protocols 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. Therefore, a team of specialists from SUAI and JSC “ISS” started to develop a new STP-ISS transport protocol that will meet industry requirements.

STP-ISS revision 1 was described in [10]. Since that time STP-ISS significantly evolved and changed, we actively work on the development and updating of STP-ISS mechanisms. This research led to the evolution of the specification. Current paper gives an overview of the second revision of the STP-ISS transport protocol. The roadmap for the STP-ISS development project is shown in Fig. 1.

II. Differences Between Revisions of STP-ISS Transport Protocol

There are four additional mechanisms are included into the second revision of STP-ISS. These are scheduling quality of service, connection-oriented data transmission, flow control and duplicate control commands detection.

As we mentioned above the STP-ISS protocol has two revisions. The first is much simpler and compact, but the second one is more powerful. Nevertheless, the backward compatibility for these revisions would be provided. Table I gives a comparison of both revisions.

TABLE I. Comparison of STP-ISS revisions

Mechanism	Rev.1	Rev. 2
Priority QoS	√	√
Guaranteed QoS	√	√
Best effort QoS	√	√
Scheduling QoS	–	√
Connectionless data transmission	√	√
Connection-oriented data transmission	–	√
CRC16 check	√	√
Reset & Flush	√	√
Packet lifetime timer	√	√
Protocol configuration possibility	√	√
Maximum data length	2 Kb	64 Kb
Flow Control	–	√
Duplicate control commands detection	–	√

ISSN 2305-7254

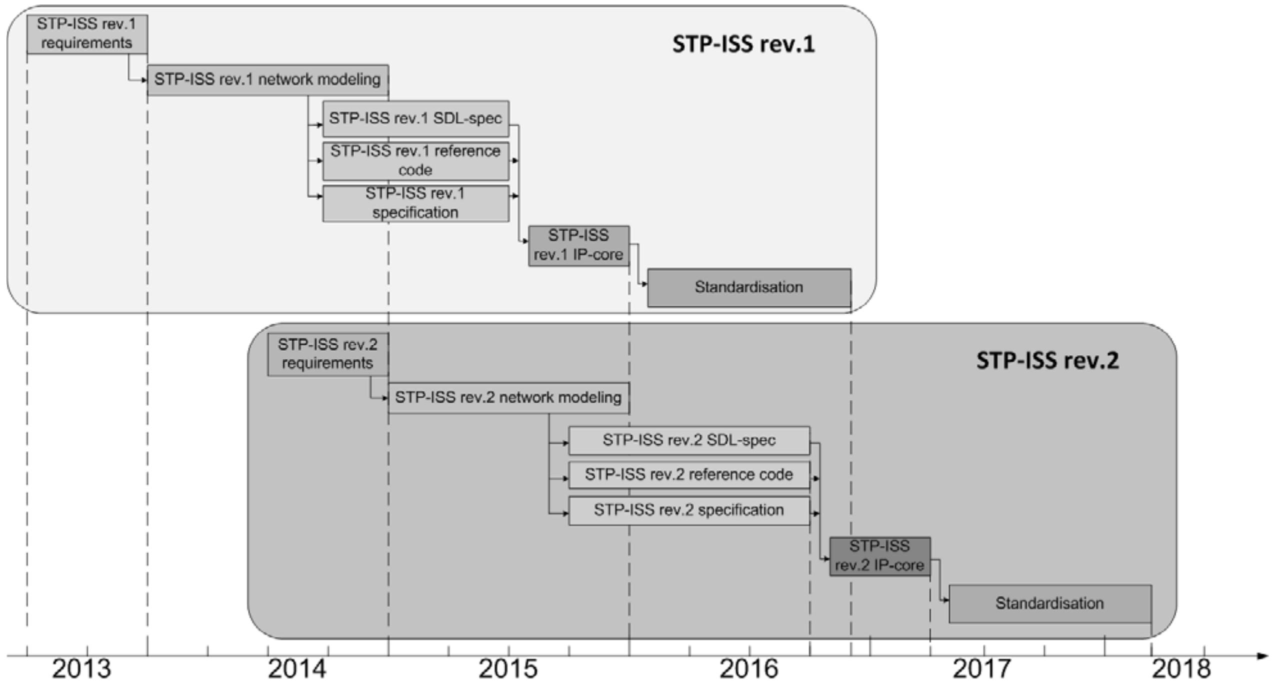

Fig. 1. STP-ISS development roadmap

III. STP-ISS REVISION 2 GENERAL DESCRIPTION

STP-ISS is a transport layer protocol that describes informational and logic interaction between on-board devices, packets' formats and packet transmission rules for SpaceWire networks. The on-board software performs functions of Session, Presentation and Application layers according to the OSI model [11]. 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. 2.

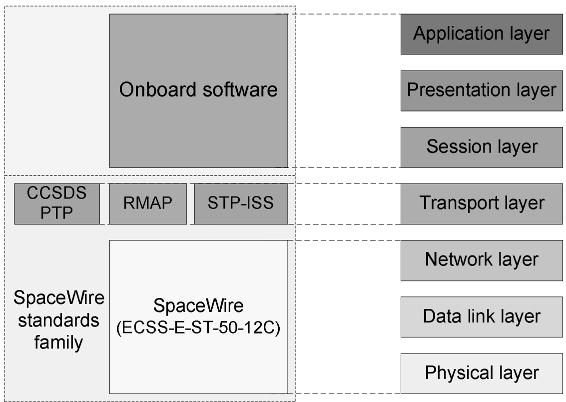

Fig. 2. The STP-ISS protocol and OSI model

A. STP-ISS interfaces

There are three interfaces for interaction between the STP-ISS and Applications: Data Interface, Configuration Interface and Control Codes Interface. At the bottom STP-ISS has two interfaces for interconnection with the SpaceWire layers: SpaceWire packets interface and Control Codes Interface (see Fig. 3).

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 a remote node by encapsulation into SpaceWire packets.

The Configuration Interface provides means for the STP-ISS configuration parameters change, for transmission of status information, reset commands and connection establishment.

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

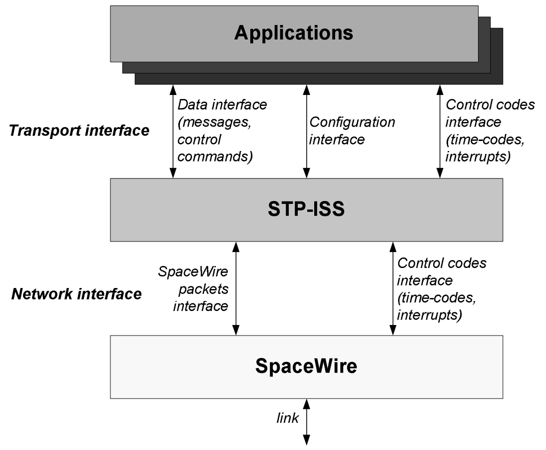

Fig. 3. STP-ISS interfaces

B. STP-ISS application messages

One of the main tasks of the STP-ISS transport protocol is to provide transmission of messages from Applications to remote nodes of SpaceWire networks. A 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 [12] at the transport layer (see Fig. 4).

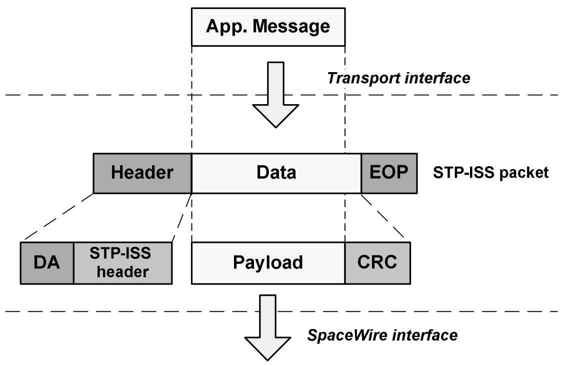

Fig. 4. STP-ISS encapsulation of a message into a SpaceWire packet

Length of each message data block should be not less than 1 byte and should not exceed 2048 bytes for the connectionless data transmission, and 64 Kbytes maximum for the connection-oriented data transmission. Segmentation of messages is done by the Application Layer. A STP-ISS packet may have a secondary header, which could be used by the Application layer to transmit some service information, for example, for assembling messages.

STP-ISS provides the reliable data transmission by using CRC-16 for protection of payload and packet header and for errors detection [13]. 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. 5).

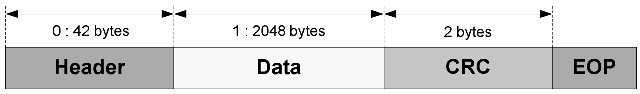

Fig. 5. STP-ISS data packet format (for connectionless transmission)

C. STP-ISS lifetime timers

STP-ISS protocol has a special packet lifetime timer, which counts the time, when the packet is still relevant 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.

D. 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 should not be less than the packet size. However, it is recommended to set the size in such a way, that buffer could be able to store two control command packets or two data packets. The resending buffers are shown in Fig. 6.

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.

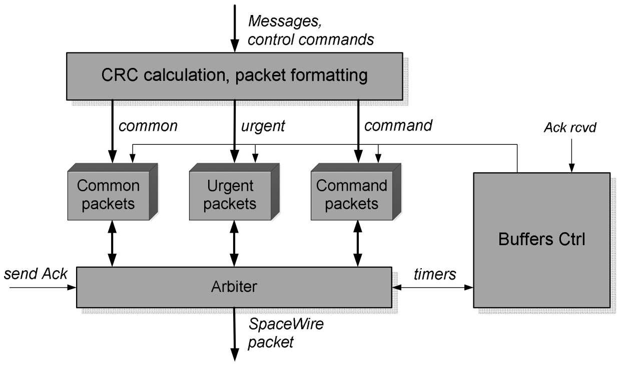

Fig. 6. STP-ISS resend buffers

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

E. Receiving buffers

In the STP-ISS second revision the receiver side of the transport protocol has two logical buffers. The first buffer is used for the connectionless data transmission, for all types of packets (control commands, common messages and urgent messages) [11].

The second buffer is used for the connection-oriented data transmission only. The receiving side should reserve required space in the buffer for each new connection. So logically the second buffer would be divided into a number of smaller buffers that is equal to the number of transport connections.

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

F. 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 perform the following actions:

- clear transmit and receive buffers,
- stop all the timers corresponding to deleted packets,
- terminate all the transport connections and delete all the related information,
- set all the configuration parameters to the default settings.

When STP-ISS gets the Flush command, it should:

- clear transmit and receive buffers,
- stop all the timers corresponding to deleted packets.

IV. STP-ISS QUALITY OF SERVICE

One of the STP-ISS benefits 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;
- scheduling quality of service.

A. 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. STP-ISS rev.2 specification supports 9 levels of priorities:

1) Acknowledgement packets and transport connection acknowledgement packets;
2) Control command packets;
3) Resend control command packets;
4) Transport connection service packets;
5) Credit synchronisation service packets;
6) Urgent data packets;
7) Resend urgent data packets;
8) Resend common data packets;
9) 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 after the current packet transmission ends.

B. 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 packet identification information. When the transmitter gets the acknowledgement, the corresponding packet should be deleted from one of the transmitter's resend buffers. All the associated with this packet timers should be stopped.

STP-ISS protocol provides the *duplicate control commands detection* in the receiver. The duplicate control command can occur in case of the loss of acknowledgement.

The duplicate control command could be detected by the combination of transmitter logical address, application number and control command number. This combination unambiguously identifies each control command.

The receiver should store the information on the last received control commands. The information on each control command should be followed by the timer. On the timer expiration the information about the corresponding control command should be deleted from the receiver.

C. 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 an STP-ISS receiver gets a best effort packet it checks the CRC and data length., In case of an error or if the packet ends with EEP, the data packet still should be sent to the Application, but with an error indication.

D. Scheduling Quality of Service

There are different scheduling mechanisms for packet-switching networks and IP networks. In general, they can be divided into two classes: reactive (or feedback control schemes) and proactive (or resource reservation algorithms) [14, 15]. Most of them are implemented by means of store-and-forward switches. Unfortunately, these mechanisms are not suitable for SpaceWire networks. SpaceWire switches support wormhole routing without full packet buffering (according to official standard).

Scheduling quality of service means that there is a single schedule for the whole SpaceWire network. This schedule gives an opportunity for the node to send data only during particular time-slots. The schedule, time-slot duration and a number of time-slots in an epoch are set during the configuration phase and are stored in each node. An epoch has a constant number of time-slots. The schedule table describes one epoch. Each epoch consists of the same number of time-slots.

The time-slot timer counts duration of the current time-slot for a particular node. A new time-slot begins if a time-slot timer expired. The expiration of a time-slot timer for a last time-slot and reception of a time-code indicate the beginning of a new epoch, in which the time-slot counter will count time-slots starting from zero. When the node gets the time-code, it does not analyse the time-code number. The beginning of a new epoch is associated with the fact of the time-code receipt.

There are two cases, which can occur in the network: the next time-code is received before or after the time-slot timer for the last time-slot is expired. It means that the internal timer and the time master timer are not synchronised. Consequently, the node should start the synchronisation process. Synchronisation is performed once in an epoch. During the synchronisation the node should calculate a new value for the time-slot timer. The newly calculated value will be applied for the time-slot timer of a new epoch. The new epoch should start when the time-code is received.

There are *K* time-slots in each epoch, when the time-code is recognized as relevant. These time-slots are called Time-code relevancy window. If a time-code is received before the last *K/2* time-slots of the epoch, or after the first *K/2* time-slots of the epoch, then this time-code is considered as irrelevant and synchronisation should not be performed (see Fig. 7).

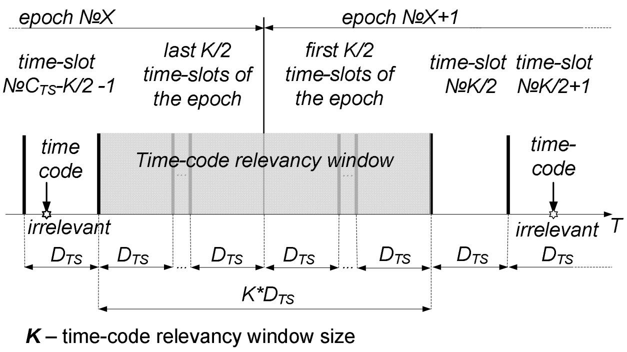

Fig. 7. An example of irrelevant time-code receiving

If the time-slot timer for a last time-slot expires simultaneously with the time-code reception, then there is no need to correct the epoch timer value.

The STP-ISS protocol should count the number of received irrelevant time-codes. Reception of three irrelevant time-codes in three consecutive epochs in a row means that the internal time-slot timer and the time master are significantly asynchronous. In this situation we also need to synchronise with the time master. Reception of the third irrelevant time-code should determine the beginning of a new epoch. The node should terminate the time-slot timer and wait for reception of the next time-code. In this new epoch the node should not send data until reception of the next time-code. After reception of the time-code the node should update the time-slot duration value and then continue data transmission according to the schedule.

We made a comparison of the modelling results for the cases, when we use STP-ISS scheduling mechanism and when do not use it. The SystemC model is represented by a SpaceWire network consisting of nodes and routers. The same topology is depicted in Fig. 11. We chose three nodes to be receivers of the 1 Kb packets from three groups of nodes. That was done to force the nodes to compete for the channels. Simulation of the model lasted for 64 milliseconds with data transmission speed 200 Mbit/s. Time-slot duration in this model is equal to 500 microseconds. The modelling results for the transmitted packets distribution in a network are shown in Fig. 8. STP-ISS scheduling mechanism provides data transmission with uniform distribution. The detailed description of the mechanism and the scientific research that lead to these results is described in [16].

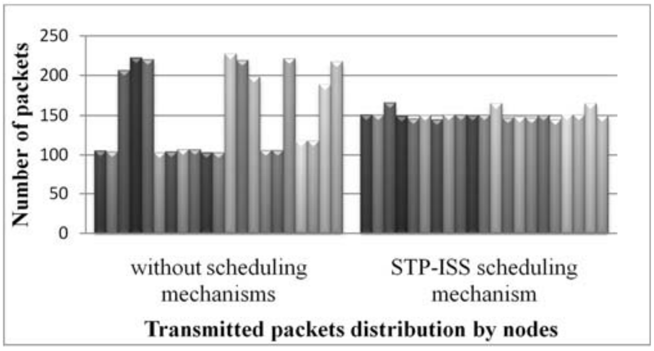

Fig. 8. STP-ISS scheduling mechanism simulation results

Current scheduling mechanism has a number of advantages:

1) it prevents conflicts of network resources usage;
2) it decreases the number of time-codes in the network;
3) it allows to increase network bandwidth;
4) it provides uniform data transmission;
5) it gives flexibility in schedule creation.

V. Connection-oriented data transmission

A. STP-ISS connection-oriented transmission description

Connection-oriented data transmission gives an ability to transmit large sized data with minimum overheads. Only urgent or common messages could be transmitted over a transport connection. Maximum number of transport connections should not be more than 8 per one direction. Each transport connection is unidirectional: it connects the transmitter of the initiator node and receiver of the remote node.

An application, which needs to transmit or receive a large portion of data, should initiate the transport connection establishment. The maximum size of data, which could be transmitted over the transport connection in a packet, is 64 Kbytes. The transport connection establishment is performed by means of classical three-phase handshake [11, 17] (see Fig. 9).

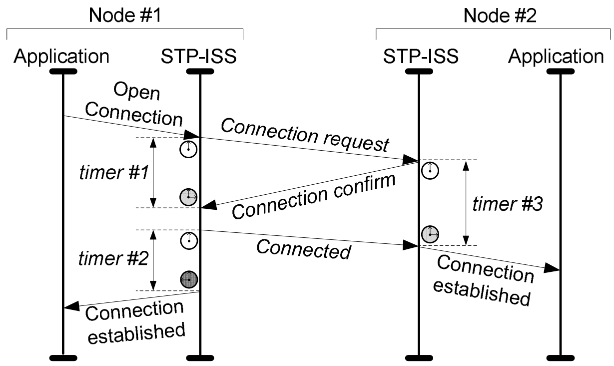

Fig. 9. An example of transport connection establishment

During the connection establishment the following connection parameters could be set:

- transport connection number;
- type of the transmitted data;
- data transmission direction (the initiator can choose, which side of the connection would send the data);
- Guaranteed or Best effort quality of service;
- maximum size of a data packet for the connection;
- space reservation in the receiving buffer (counted in number of messages).

The remote node checks if it is able to establish the connection and responds with the confirmation or reject. After the successful connection establishment the node, which was chosen to be a transmitter, starts to send data.

For each transport connection receiver and transmitter has the standby timer. This timer counts the time of waiting for the next data or service packet transmitted over the connection. On standby timer expiration the transport connection should be closed and the Application should be indicated about it.

The connection closing is initiated by the Application, which was the initiator of the current connection establishment. It is also performed using the three-phase handshake connection close.

B. Flow control mechanism

The flow control is performed by sending of the information about the available free space in the receiving buffer. This mechanism is applied only for the transport connections with the guaranteed quality of service.

The receiving node during the transport connection establishment should reserve the requested buffer space and keep counting it. Each transport connection has a Free Credits counter and a Used Space counter. The information on available free space in the receiving buffer should be sent in the transport connection acknowledgement packet. In case of the free space change after sending of a transport connection acknowledgement (e.g. an Application reads the data from buffer), receiver sends a transport connection acknowledgement with a special flag, which indicates that it is not an acknowledgment, but the new credit counter information.

In turn, the transmitting node counts the amount of the data that could be sent using the credit counters for each transport connection.

The loss or corruption of an acknowledgement can cause a loss of the information on new free credits in the receiver. To avoid this STP-ISS provides the credit synchronisation mechanism. The transmitting node starts the credit synchronisation after each N packets sent. For this purpose the transmitting node sends credits request packet. The receiving node gets it and answers with the credits synchronisation response packet with the current value of free buffer space. Then the transmitting node should check if its local free buffer space counter is valid. If it is not valid – the local free buffer space counter value should be changed to the received value.

VI. 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. 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 on-board device;
- emergency recovery.

The current STP-ISS specification describes 13 configuration parameters.

The configuration flexibility gives an opportunity to switch off or even not to implement some mechanisms of STP-ISS. It has a variety of implementation profiles. This can be beneficial for different types of the on-board equipment. For example, a simple sensor may not need the complex mechanisms of guaranteed QoS or scheduling. However, this mechanism would be very useful for the intelligent nodes of the SpaceWire network.

There are some mechanisms that should be implemented as mandatory. They are:

- Priority QoS (at least 1 priority);
- Best effort QoS;
- CRC16;
- Transmit buffer (at least for one type of messages);
- Receive buffer (at least for one type of messages).

The other mechanisms can be considered as extensions and could be optionally implemented in different combinations. In particular: Guaranteed QoS; Scheduling QoS; Connection-oriented data transmission; Duplicate control commands detection.

The next section describes the STP-ISS protocol application for the on-board networks.

VII. STP-ISS APPLICATION FOR ON-BOARD NETWORKS

STP-ISS protocol is developed for the on-board SpaceWire networks and it is planned for the future use in spacecrafts. STP-ISS does not depend on a network topology; it allows using all kinds of addressing types that are specified for SpaceWire.

Fig. 10 shows an example of the on-board network for a small-size satellite. Dotted lines show the information flows from sensors to the other parts of the satellite. However, STP-ISS also can be used for the much more complex networks. The maximum number of the nodes is regulated by the SpaceWire technology.

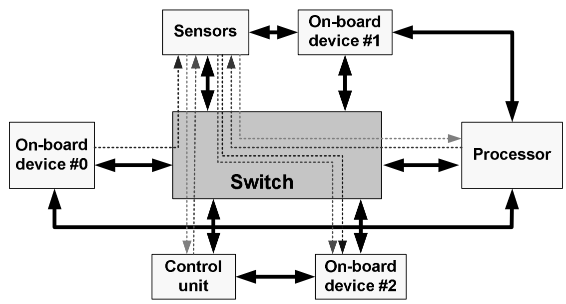

Fig. 10. An example of the on-board network topology

In addition, a network could be divided into several regions. Fig. 11 shows an example of the on-board network with several regions.

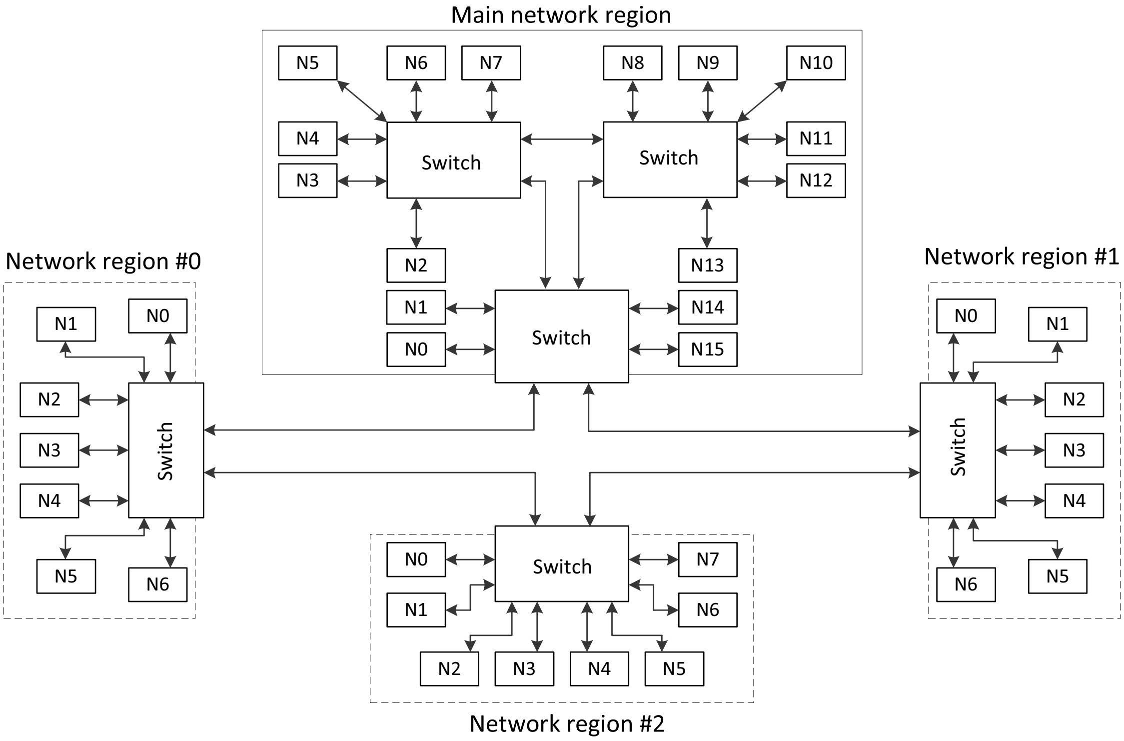

Fig. 11. An example of the on-board network topology with regions

VIII. Conclusion

The current paper gave a description of our experience in the field of development of the new transport protocol for the SpaceWire networks. We gave a roadmap of the project, compared the new revision of STP-ISS transport protocol with its first revision. Then we described in details all major internal mechanisms of the protocol, in particular we presented the principles of the new STP-ISS rev. 2 mechanisms: scheduling quality of service, connection-oriented data transmission, flow control and duplicate control commands detection. The STP-ISS protocol rev. 2 satisfies all the technical requirements from the industry.

The new STP-ISS protocol is developed to operate over SpaceWire networks in spacecrafts. To give the basic notion of a structure of the on-board network the paper presents two estimated network topologies.

Both revisions of the STP-ISS protocol have been verified and tested by the following types of models: SDL specification [18], reference-code in C++ and SystemC [19, 20] network modelling [21]. As a result of the modeling work we have fixed a number of issues in the specification [10].

The next stage of our work is the IP core for STP-ISS revision 2. Then we plan to perform the joint software/hardware testing for both revisions. Finally, when the equipment with STP-ISS support is ready, the ground-based test operation would be done.

The first on-board testing in a real satellite for the STP-ISS revision 1 equipment is planned for the 2018.

Acknowledgement

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation under the contract RFMEFI57814X0022.

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] Y. Sheynin, V. Olenev, I. Lavrovskaya, I. Korobkov, S. Kochura, S. Openko, D. Dymov "STP-ISS Transport Protocol Overview and Modeling", *Proceedings of 16th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program.* Oulu: University of Oulu, 2014. pp.185-191.

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

[12] P. Walker, B. Cook "SpaceWire: Key principles brought out from 40 year history", *20th Annual AIAA/USU Conference on Small Satellites*, Utah State University, 2006.

[13] Koopman, P., Chakravarty, T. "Cyclic redundancy code (CRC) polynomial selection for embedded networks". *DSN '04 Proceedings of the 2004 International Conference on Dependable Systems and Networks*, IEEE Computer Society, 2004. pp. 145-154.

[14] A. La Piana, I. Ferraro, C. Beoni, "Queuing, scheduling and traffic shaping over IP networks". *Vehicular Technology Conference VTC 2004.* IEEE 59th, Vol. 4, 2004. pp. 2322-2326.

[15] H. Zhang. "Service disciplines for guaranteed performance service in packet-switching networks"*. Proceedings of the IEEE, Vol. 83, No.10.* IEEE, 1995. pp. 1374-1396.

[16] V. Olenev, E. Podgornova, I. Lavrovskaya, I. Korobkov, N. Matveeva. "Development of the transport layer scheduling mechanism for the onboard SpaceWire networks". *Proceedings of 16th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program.* Oulu: University of Oulu, 2014. pp. 164-170.

[17] Tomlinson, R.S., "Selecting Sequence Numbers", *Proceedings of the ACM SIGCOMM/SIGOPS Interprocess Communication Workshop, and ACM Operating Systems Review*, Vol. 9, No. 3, July 1975, Association for Computing Machinery, New York, 1975.

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

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

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

[21] V. Olenev, I. Lavrovskaya, P. Morozkin, A. Rabin, S. Balandin, M. Gillet. " Co-Modeling of Embedded Networks Using SystemC and SDL". International Journal of Embedded and Real-Time Communication Systems (JERTCS), #2(1). Tampere: IGI Global, 2011. pp. 24-49.