Analysis of the Transport Protocol Requirements for the SpaceWire On-board Networks of Spacecrafts

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

Dmitry Dymov
JSC "Academician M.F. Reshetnev" Information Satellite Systems"
Zheleznogorsk, Russia
dymovdv@mail.ru

***Abstract*—The paper gives an overview and analysis of SpaceWire oriented transport protocols. Also this paper considers the "Academician M.F. Reshetnev" Information Satellite Systems" general requirements for the Transport protocol to operate over the SpaceWire network technology. Finally, we propose preliminary solutions for some Transport protocol mechanisms.**

I. INTRODUCTION

SpaceWire is a data-handling network for spacecraft which combines simple, low-cost implementation, with high performance and architectural flexibility [1]. Its advantages over competing technologies have been demonstrated by its rapid take up by the normally conservative international space agencies and space industry. SpaceWire is now being used on more than 30 high profile missions and by all of the major space agencies and space industry across the world [2].

SpaceWire is primarily intended for data-handling applications but does not address avionics and other applications where responsiveness, robustness, determinism and durability are essential requirements. Mil-Std 1553 [3, 4] has long been the communications bus of choice for spacecraft avionics. 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. On-board payload data-handling is now dominated by the SpaceWire standard. The need in smaller spacecraft, planetary landers, etc., for integrated avionics and data-handling networks has raised the possibility of using SpaceWire for avionics applications.

SpaceWire is a network technology which does not provide transport layer services. Current Russian space industry demands a Transport protocol running over SpaceWire which will provide reliability, guaranteed services and determinism.

II. SPACE ORIENTED TRANSPORT PROTOCOLS REVIEW

Nowadays there is a number of transport protocols intended to operate over SpaceWire. They are: RMAP, CCSDS PTP, STUP, JRDDP and STP (see Fig. 1). Each of them is intended to solve its particular tasks.

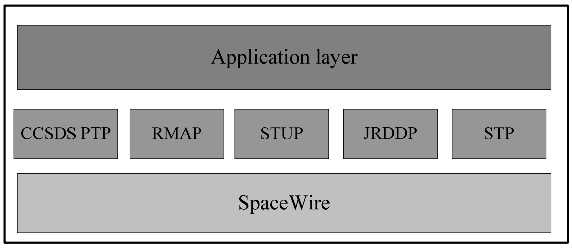

Fig. 1. Transport protocols over SpaceWire

This section will give an overview of these protocols in order to elaborate if any of them meet the Russian space industry requirements.

A. *Remote Memory Access Protocol*

The Remote Memory Access Protocol (RMAP) has been designed to support a wide range of SpaceWire applications. Its primary purposes however are to configure a SpaceWire network, to control SpaceWire nodes and to gather data and status information from those nodes. RMAP can operate alongside other communication protocols running over SpaceWire [5].

RMAP can be used for the following purposes:

- To configure SpaceWire routing switches, setting their operating parameters and routing table information.
- To monitor the status of those routing switches. RMAP can be used to configure and read the status of nodes on the SpaceWire network.

- For simple SpaceWire units without an embedded processor, to set application configuration registers, to read status information and to read from or write data to memory in the unit.
- For intelligent SpaceWire units to provide the basis for a wide range of communication services. Configuration, status gathering and data transfer to and from memory or mailboxes can be supported [5].

The RMAP protocol can be described by its following general features:

- RMAP is a connectionless transport protocol;
- supports path, logical and regional addressing;
- write commands can be acknowledged or not acknowledged, verified and not verified;
- provides a means of reading and writing of data into the memory by just one command (read-modify-write command);
- no timeouts;
- no flow control.

RMAP defines three types of commands:

1) Write commands. The write command provides means for one node (the initiator) to write zero or more bytes of data into a specified area of memory in another node (the target) on a SpaceWire network. Write commands can be acknowledged or not acknowledged by the target when they have been received correctly. Write commands can perform the write operation after verifying that the data has been transferred to the target without error, or it can write the data without verification. Verification on the data can be performed only by buffering in the target to store the data while it is being verified, before it is written. Larger amounts of data can be written but without verification prior to writing. Verification in this case is done after the data has been written. It is recommended to perform verification when writing control and configuration registers [5].

2) Read commands. The read command provides a means for one node, the initiator, to read zero or more bytes of data from a specified area of memory in another node, the target on a SpaceWire network. The data read is returned in a reply packet which normally goes back to the initiator. If the data could not be read by some reason then the reply packet to the initiator should contain the error code.

3) Read-modify-write commands. The read-modify-write command provides a means for one node, the initiator, to read a memory location in another node, the target, modify the value read in some way and then write the new value back to the same memory location. This command can contain a mask specifying which register bits should be written. The original value read from memory is returned in a reply packet to the initiator. This reply is also used for indication of the operation success.

RMAP protocol provides guaranteed service in an acknowledged mode and best effort service in a non-acknowledged mode.

B. CCSDS Packet Transfer Protocol

CCSDS Packet Transfer Protocol (CCSDS PTP) – is a packet transfer protocol which encapsulates a CCSDS Space Packet into a SpaceWire Packet, transfers it from an initiator to a target across a SpaceWire network, extracts it from the SpaceWire packet and passes it to the target user application [6].

PTP provides a unidirectional data transfer service between source and target user applications over SpaceWire network [6].

The CCSDS PTP protocol can be described by its following general features:

- connectionless protocol;
- user may request data transfer at any time;
- variable or fixed packet length (minimal length is 7 bytes, maximal – 65542 bytes);
- unidirectional data transfer without acknowledgments;
- no data retransmission mechanism;
- no packet verification (it's a user application functionality) [6].
- CCSDS PTP doesn't provide any mechanisms for guaranteeing a particular quality of service [6].

C. Serial Transfer Universal Protocol

Serial Transfer Universal Protocol (STUP) is intended for data transfer over the SpaceWire network. Its main feature is a minimized complexity [7].

The general features of the STUP protocol are:

- connectionless protocol;
- easy to implement protocol (minimized complexity);
- just 2 types of commands: write and read.

STUP protocol does not provide any mechanisms for guaranteeing quality of service except best effort [7]. However, STUP commands have checksum fields for verification of received data.

D. *Joint Architecture Standard Reliable Data Delivery Protocol*

Joint Architecture Standard Reliable Data Delivery (JRDDP) is a protocol which provides reliable data transmission. It uses the lower-level SpaceWire data link layer to provide reliable packet delivery services to one or more higher-level host application processes [8].

The JRDDP protocol has the following main features:

- connection-oriented protocol;
- multiple logical connections;
- reliable data delivery;
- detection of missing packets;
- out-of-sequence packet reordering;
- buffer fragmentation and reassembly [8].

JRDDP defines the following packet types:

- application data;
- acknowledge;
- open/reset command;
- close command;
- urgent.

JRDDP provides two types of quality of service: priority and best-effort.

According to JRDDP specification the data flows should have the following priorities:

- acknowledgment packets (transmits first);
- control packets;
- urgent packets;
- retransmit packets;
- data packets (transmits last).

The best-effort QoS is optionally used for urgent messages delivery such as time broadcasts, messages with exceptions and errors control, meta-messages, etc. [8].

JRDDP protocol provides fault detection and fault tolerance by means of CRC checksum and packet sequence numbering. Moreover, it uses timeouts for detection of missing and duplicate packets and acknowledgements for indication of successful packets delivery.

E. *Streaming Transport Protocol*

Streaming Transport Protocol (STP) is developed for streaming data transmission over SpaceWire network. This protocol also supports simultaneous transmission of multiple coherent data flows [9].

The STP protocol is oriented for asymmetric establishment of transport connection: on the one side there is a host (master), and the slave device on the other side. The host device is an initiator of a transaction session. The master performs the connection establishment, configuration of connection parameters and packets flow control [9].

The STP protocol can be described by its following general features:

- connection-oriented protocol;
- reliable handshake for connection establishment and teardown (3-way handshake);
- asymmetric connection (data transmission is performed from slave to host device);
- multi-streaming (up to 65535 connections);
- fixed length of transmitted data;
- periodical data transfer in specified time period in accordance with the configuration parameters and during the whole duration of the connection;
- data delivery without acknowledgements and retransmission;
- data flow control.

The STP specification defines data and command packets.

STP was designed for streaming data transmission over a SpaceWire network. A target device does not acknowledge arrived packets. Thereby, STP provides best-effort quality of service.

Likewise, STP supports flow control by means mechanisms of enabling and disabling data transmission via a particular transport connection and periodical packet sending.

STP protocol uses the following mechanisms to provide fault detection and fault tolerance:

- packet fields verification, header and payload CRCs;
- timeouts mechanism;
- terminal node status monitoring procedure (status command sending).

F. *Protocols comparison*

General features of each overviewed protocol are given in the Table I.

Nowadays, there is no transport protocol over SpaceWire which can provide different types of quality of service and guaranteed data delivery. Currently existing protocols which were designed specifically for SpaceWire, such as RMAP, CCSDS PTP, STUP and STP, are not dedicated for the stated tasks. RMAP provides rich means for switch configuration and monitoring. The CCSDS PTP protocol is primarily intended for space packets encapsulation into the SpaceWire packets for their further transmission over the network. STUP and STP protocols, 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 it is not guaranteed.

TABLE I. PROTOCOLS COMPARISON

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

Therefore, there is no SpaceWire oriented transport protocol providing reliability, guaranteed services and scheduling. For these reasons a new Transport protocol is planned to be developed.

III. GENERAL REQUIREMENTS FOR THE TRANSPORT PROTOCOL

To develop a Transport protocol conforming current space industry demands Saint-Petersburg State University of Aerospace Instrumentation in collaboration with JSC "Academician M.F. Reshetnev" Information Satellite Systems" [10] elaborated the requirements to the new Transport protocol. These requirements were worked out in the scope of the contract between JSC ISS and SUAI and are for the first time presented in this paper. This section gives a list of the main requirements.

A. Transport interface

The Transport layer protocol should provide transmission of the following general data flows passing from the Application layer:

- control commands;
- application process messages;
- time codes;
- interrupt codes and interrupt acknowledge codes.

B. Segmentation

Segmentation of large messages should be performed on the Application layer. The target segments with the additional service information should be passed from the Application layer to the Transport layer.

C. Data flows and priorities

Each data flow should have its particular priority. The data flows should have the following precedence:

1. Control commands – the highest;
2. Urgent messages (in the transmission order from the Application layer);
3. Common messages (in the transmission order from the Application layer) – the lowest.

D. Buffering on the transmitter side

Transport protocol should comprise a separate logical buffer for each data flow priority.

E. Quality of service

The SpaceWire standard does not provide guaranteed services. Therefore, the target Transport protocol should provide an additional fault detection level over the SpaceWire connection by means of the following mechanisms:

- CRC checksum;
- packet sequence numbers;
- acknowledgements of the successful packet receipt;
- detection of lost packets by timeouts.

Each transport data flow is characterized by particular features and, consequently, requires its particular quality of service. The summary of required data flow characteristics is given in the Table II. According to the table the priority quality of service is required by all data flows. Control commands and urgent messages should be delivered reliably. In turn, common messages can be transmitted reliably or just in a best effort way.

Basing on the table, the intensity of generation data of a particular flow depends on its length and type.

IV. POSSIBLE SOLUTIONS

Taking into account the given above requirements and the transport protocol review we propose preliminary solutions for some protocol mechanisms. The following subsections deal with such aspects as data formats, CRC, buffering and quality of service.

A. Formats

According to the general requirements there are three types of incoming messages from the Application layer which should be handled by the Transport protocol: control commands, urgent messages and common messages.

The format of the control command should be different from the urgent and common message, because it has fixed data length and it is smaller than in other messages.

The incoming messages should be divided into segments on the Application layer, so the Transport layer should get from the application not only the message itself, but also the number of a segment. In this case the packet should have a special field for a secondary header. This secondary header comprises a segment number and an end of message flag. The segment number is used by the remote side of the transmission to assemble the original message. The end of message flag is used to detect the last segment of the message.

TABLE II. REQUIRED QUALITY OF SERVICE OF TRANSPORT PROTOCOL DATA FLOWS

№	Data flow	Length	Intensity	Latency	Quality of Service	Priority	Acknowledgement
1	Control commands	16 bits	≥ 1 ms	≤ 0,5 ms	priority, scheduling	1	Yes
2	Urgent messages	4 bytes 1 Kbyte 64 Kbytes	≥ 0,2 ms ≥ 5 ms ≥ 250 ms	≤ 0,25 ms ≤ 0,5 ms ≤ 40 ms	priority, scheduling, guaranteed	2	Yes
3	Common messages	4 bytes 1 Kbytes 64 Kbytes	≥ 0,2 ms ≥ 5 ms ≥ 250 ms	≤ 1 ms ≤ 1 ms ≤ 80 ms	priority, scheduling, guaranteed, best effort	3	Yes / No
4	Time codes	6 bits	≥ 60 s	≤ 0,1 ms	priority	0	No
5	Interrupts, interrupt acknowledges	5+1 bits	≥ 5 ms	≤ 0,1 ms	priority	0	Yes / No

B. CRC checksum

The designed Transport protocol shall use CRC checksum for the reliable data delivery. It is a widely used mechanism, which is successfully applied for the most of the overviewed transport protocols.

We propose to use the CRC checksum like it is done in RMAP protocol. The packet header would be covered with the CRC-8, which is enough for the current header format and length. The data field length would be limited in order to use the CRC-16. So, the maximum data field length would be 2048 bytes.

C. Buffering

According to the general requirements the transmitter side of the Transport protocol should contain the buffer for each type of the incoming messages:

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

Each packet should have an identification number. This will give an ability to send it back in the acknowledgement and so indicate the successful transmission of the packet.

The transmitted packet is stored in the buffer until the receiver gets the acknowledgement with the same identification number, or a lifetime timer expires. In case of need this will give an ability to resend the packet.

When the packet is deleted from the buffer, the Application layer should be indicated about this. The indication gives information on the packet successful delivery, when the packet is deleted from the buffer after the acknowledgement, or an unsuccessful delivery, when the packet is deleted after the expiration of the lifetime timer. Depending on this the Application layer can control the transmission of the data and resend the data, which has not been sent successfully.

D. Quality of service

The proposed Transport protocol will provide the following quality of service types: priority transmission, guaranteed data delivery, scheduling and best effort. These four types of the quality of service should be enough for the required functionality of the protocol.

Transport protocol has 7 levels of the priority (the lower number – the higher priority):

1. Acknowledgment packets;
2. Control command packets;
3. Resent control command packets;
4. Urgent message packets;
5. Resent urgent message packets;
6. Resent common message packets;
7. Common message packets.

The acknowledgement packets have the highest priority to extremely fast acknowledge the successful packet delivery and empty the place in the buffer.

Control command packets have the higher priority than resent control command packets, because there is a high possibility that after the resending the control command will not be actual any more for the remote device. In this case it would be better to send a new control command than to resent the old one.

Urgent message packets have the higher priority than resent urgent message packets due to the same reason.

Herewith the common message packets have the lower priority than resent common message packets. Otherwise, if the device would have the intensive incoming data stream from the Application layer, then common message packets will wait for the resending for too long and there is a high possibility that outgoing common messages buffer will be overflowed. So the priority is changed and resent common message packets have the higher priority.

E. Configuration parameters

There are a number of configuration parameters provided for the designed Transport protocol. These parameters should be set during the device configuration stage. It is in order for the proper operation and flexible use of the protocol.

For example, “Guaranteed data delivery” and “Scheduling” parameters give an ability to switch off the corresponding quality of service type and operate in a limited mode. Such an ability could be used for the small-sized spacecrafts.

V. CONCLUSION

The paper gives an overview and analysis of SpaceWire oriented transport layer protocols against the requirements of space industrial company "Academician M.F. Reshetnev" Information Satellite Systems". This analysis showed that there is no such transport protocol for SpaceWire network technology providing reliability, guaranteed services and scheduling, which are the main requirements of space industry. Therefore, such protocol shall be developed and our current work direction is focused on this task. In our paper we proposed key solutions for several Transport protocol mechanisms: data formats, CRC checksum, buffering on the transmitter side, quality of service and configuration facilities. These proposals can become a basis for a new Transport protocol after their evaluation, verification and validation by mathematical, program modeling and hardware testing.

REFERENCES

[1] ECSS, *SpaceWire — Links, nodes, routers and networks (ECSS-E-ST-50-12C)*. Noordwijk: ESA Requirements and Standards Division, July 2008.

[2] Steve Parkes, “SpaceWire-RT (SpWRT)”, *Let’s embrace space*, Vol. II, Luxembourg: Publications Office of the European Union, 2012.

[3] MIL-STD-1553B, *Interface Standard for Digital Time Division Command/Response Multiplex Data Bus*, Notice 4, 15 January 1996.

[4] MIL-STD-1553 Tutorial, *AIM Worldwide*, November, 2010, Web: http://www.aim-online.com/pdf/OVW1553.PDF

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

[6] *Standard ECSS-E-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] Mike Gardner, Richard Hunt, Justin Enderle, Daniel Gallegos, John Eldridge, and Jim Daniels, *Joint Architecture System Reliable Data Delivery Protocol (JRDDP)*. Sandia National Laboratories, Albuquerque, New Mexico, May, 2011.

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

[10] JSC "Academician M.F. Reshetnev" Information Satellite Systems" official website, Web: http://www.iss-reshetnev.com