Queuing System for the SpaceFibre Standard

Ilya Korobkov
Saint-Petersburg State University of Aerospace Instrumentation
Saint Petersburg, Russia
ilya.korobkov@guap.ru

***Abstract*—The streaming data over onboard spacecraft and aircraft networks is considered in the paper. The SpaceFibre spacecraft standard could be successfully used for streaming. It is proven by the proposed comparison of SpaceFibre with other network standards. While investigating and analyzing SpaceFibre for streaming, the limitation of the SpaceFibre QoS mechanisms was found. The solution was also proposed. To evaluate its benefits the mathematical approach to quantity calculation of transmitted streaming data via SpaceFibre was develop. The paper is concerned with this approach and the mathematical model of the streaming data transfer over SpaceFibre virtual channels.**

I. Introduction

Modern spacecrafts and aircrafts consist of many onboard systems and sensors. A set of system/sensors and switches are onboard data-handling network. Communications and data exchange between onboard systems could be cyclic [1] and streaming [2], [3], [4], [5]. Large number of high rates sensor sampling and onboard video camera produce high intensive data streams that generate high-rate coherent streams.

A. *Streaming traffic*

There are several definitions of streaming traffic [4]:

- Traffic type characterized by viewing and/or listening for information as new information becomes available.
- Streaming data transfer is a way of transferring real-time or buffered data such as sound, video, documents or photos through the networks with acceptable Quality of Service. Receiving system can start playback or display data before receiving full information.
- Streaming traffic is the uniform data stream with a constant bit rate.

The third definition is more general, but it is the most representative for streaming traffic. All over the world the streaming traffic is used for the next data transfer tasks:

- Audio/Video translations: end-to-end communications; multicast translations; broadcast teleconferences;
- IF images streams;
- High-rate informative streams: telemetry streams, sensor array streams (SAR), etc.

There are several main features of streaming traffic: fixed packet size (no wide range of sizes); periodic packet issue with stable intensity; tolerance to single and sporadic corruption. Streaming traffic allows to predict buffer size and optimize streamline sender/receiver equipment [4].

B. *Quality of Service*

Every data stream in onboard spacecraft network is required a particular Quality of Service (QoS) [2], [4], [5]. Mission critical data, science data and live streaming interactive video specify special requirements for data delivery speed, latency and jitter [4], [5], [6]. Usually the use of various types of QoS adjusted to the traffic characteristics and the requirements of mission specific applications is needed. The most common QoS are the Best Effort, Guaranteed, Priority, Scheduling and Bandwidth Reservation [5].

1) The Best Effort QoS: It is used if there are no noise and fluctuations of available bandwidth in a channel. In this case data transfer could be transmitted without acknowledgements, retransmission, timeouts, etc. The data transfer may be connection-oriented, because there is no need to add, for example, a source destination address to every sent data packet. The connection between two end-users should be established instead. After that end-users are informed about communication between them. Hence, there is no need to transfer general information in packet: length of address, source address, etc. Data packets transfer has minimal overheads. The Best-Effort QoS is the most effective for the broadband traffic, responsive to time delays and loss tolerance. Also it can be applied to the narrowband traffic.

2) The Guaranteed QoS: It is used if there is noise or transmitted data is a critical information. To provide this QoS the following mechanisms are used: selective acknowledgments, immediate or postponed data retransmission, numbering of packets and timeouts for confirmation. This QoS is most effective for the narrowband traffic, not responsive to time delays and sensitive to losses. Also it can be applied to the broadband. The combinations of different mechanisms of the Guaranteed QoS give an opportunity to use it for transfering streaming traffics providing the guaranteed delivery or a speed/time delay of packet delivery.

3) The Priority Transmission QoS: It is used when there is a need to send urgent information bypassing the current data streams. For example, there is a stream video translation and there is a need to send urgent commands or warnings of some alerts. The high-priority packet will be immediately created and sent to, for example, onboard control complex. Then video translation will be proceeded. The transmission on priorities can be applied to any streaming traffic.

4) The Scheduling QoS: It is applied to networks for the purpose of the conflict resolution. The conflict appears when two or more nodes send data to one destination device at the same time. The Scheduling quality of service means that there is a single schedule for the whole network. This schedule gives an opportunity for the node to send data only during particular time-slots. Thus, it prevents conflicts of a network resources usage.

5) The Bandwidth Reservation QoS: It is used for the same purpose as the Scheduling QoS. A node is allowed to transmit particular volume of data.

C. Review of the onboard network standards

The onboard spacecraft and aircraft networks are built on the various network standards. They are RapidIO, SpaceWire, SpaceFibre, ARINC-818-2, etc. To meet the streaming traffic requirements the analysis and comparison of the standard features have been done. It is based on researches [2], [4], [5], [7]. Short description of the standards is presented below in subsections.

1) SOIS – Spacecraft Onboard Interface Services: The CCSDS SOIS Area has developed a layered set of communications services for flight avionics. This set of services is intended to cover the majority of onboard communications requirements. The services have been divided into those to be provided over the onboard communications media — the so-called Subnetwork Layer services — and those supporting onboard applications—the Application Support Layer services.

2) RapidIO – Rapid Input/Output Interface: It is a high-performance packet-switched, interconnect technology. RapidIO supports messaging, read/write and cache coherency semantics. RapidIO fabrics guarantee in-order packet delivery, enabling power- and area- efficient protocol implementation in hardware. RapidIO can be used as a chip-to-chip, board-to-board, and chassis-to-chassis interconnect.

3) ARINC-818-2 – Avionics Digital Video Bus: It is a video interface and protocol standard developed for high bandwidth, low latency, uncompressed digital video transmission in avionics systems: Boeing 787, Airbus A380, A400, C-130, F18, F22, F35.

4) SpaceWire: It is a data-handling network for use on-board spacecraft, which connects together instruments, mass-memory, processors, downlink telemetry, and other on-board sub-systems. SpaceWire is simple to implement and has some specific characteristics that help it support data-handling applications in space: high-speed, low-power, simplicity, relatively low implementation cost, and architectural flexibility making it ideal for many space missions. SpaceWire provides high-speed (2 Mbits/s to 200 Mbits/s), bi-directional, full-duplex data-links, which connect together SpaceWire enabled equipment.

5) SpaceFibre: It is a very high-speed serial link designed specifically for use onboard spacecraft. It aims to complement the capabilities of the widely used SpaceWire onboard networking standard: improving the data rate by a factor of 10 (2Gbit/s), reducing the cable mass and providing galvanic isolation. Multi-laning improves the data-rate further to well over 20 Gbits/s. SpaceFibre provides a coherent quality of service mechanism able to support best effort, bandwidth reserved, scheduled and priority based qualities of service. It substantially improves the fault detection, isolation and recovery (FDIR) capability compared to SpaceWire.

Features of the each reviewed standards are given in the Table I.

TABLE I. FEATURES OF ONBOARD NETWORK STANDARDS

MECHANISMS AND FEATURES	Spacecraft Onboard Interface Services	RapidIO	ARINC-818 rev. 2	SpaceWire	SpaceFibre
Protocol Data Unit (PDU)	Depends on the transport and data link layer protocols	Data Streaming packet	Fiber Channel frame	Packet	
PDU header length, bytes		4-8	24-28	Unlimited	
Max PDU payload, bytes		64K	2112		
Function	Spacecraft	Avionics		Spacecraft	
PDU fixed size	+-	-	-	-	-
Periodical data issue	+-	-	-	-	-
Delivery without ACKs	+	+	+	+	-
Priority QoS	+	+	-	-	+
Bandwidth Reservation QoS	+	+	-	-	+
Scheduling QoS	+	-	-	-	+
Data correctness check	+	+	+	+	+
Data sequence check	Depends on the transport and data link layer protocols	+	-	-	+
Only without packet retransmission		-	+	+	-
Selective ACKs		+	-	-	+
Time stamp in PDU		-	+	-	-
Network Status (overload)	+	-	-	-	+-

According to the conducted analysis, nowadays, there is only SpaceFibre which could provide all types of Quality of Service (QoS), selective acknowledgments and network status (partly). Researches of various streaming traffics transmission over SpaceFibre networks were performed and described in [7-14]. As demonstrated in [14], there is the limitation and disadvantage of the SpaceFibre QoS mechanisms – problem of idle virtual channel. The solution was also proposed in [15]. To evaluate its benefits - how much the bandwidth of idle virtual channel will increase and how much data will be transferred via all virtual channels of SpaceFibre – it is necessary to develop the mathematical approach to quantity calculation of transmitted streaming data via SpaceFibre virtual channels. Moreover, it is essential for designers of the SpaceFibre spacecraft networks. This approach would make the process of the SpaceFibre network configuration (priorities, time-slots, etc.) much easier. The paper is concerned with the approach mentioned above. Next sections describe it.

II. APPROACH TO QUANTITY CALCULATION OF TRANSMITTED STREAMING DATA VIA SPACEFIBRE OUTPUT VIRTUAL CHANNELS

A. What is SpaceFibre, General Description

SpaceFibre (SpFi) is a very high-speed serial data-link, which is intended for use in data-handling networks for high data-rate payloads [16], [17].

1) Virtual Channels: SpaceFibre interface includes a number of input and output virtual channels. Each provides a FIFO type interface. When data packet is placed in a SpaceFibre output virtual channel it is transferred over the SpaceFibre link and placed in the same numbered input virtual channel at the other end of the link. Data from the several output virtual channels (VCs) are interleaved over the physical SpaceFibre connection. To support the interleaving, data is sent in short data frames of up to 256 SpaceWire N-chars (data bytes) each. SpaceFibre supports Priority, Bandwidth Reservation and Scheduled QoS [16], [17], [18].

2) Medium Access Controller and Precedence: There can be several VCs connected to a single data link, which compete for sending information over the link (see Fig. 1). The medium access controller (MAC) arbitrates between all the VCs requesting to send a frame. It uses the precedence of each of the requesting VCs to determine which one will be allowed to send the next data frame. VC with the highest precedence will be allowed to send the next frame. In general case, the precedence depends on Priority level and Bandwidth Credit. The last one updates every time a data segment has been sent on any virtual channel [16], [17], [18].

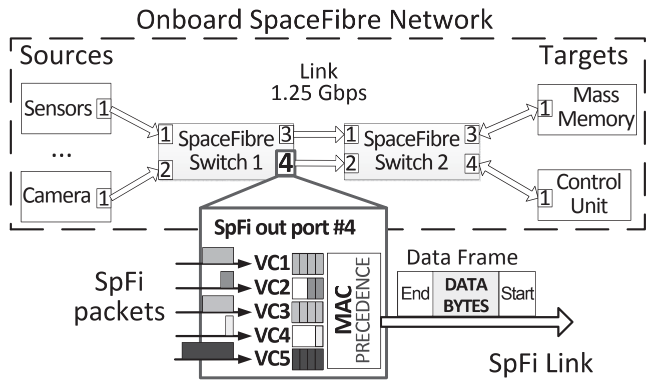

Fig. 1. Output Virtual Channels and MAC into out-port of SpaceFibre Switch

3) Bandwidth Reservation QoS: A virtual channel specifies a portion of overall link bandwidth that it wishes to reserve and expects to use. When a frame of data is sent by any virtual channel, each virtual channel computes the amount of bandwidth that it would have been permitted to send in the time interval that the last frame was sent. This information is used to specify the precedence of virtual channel [16, 17, 18].

4) Priority QoS: Each VC is assigned a priority level: 0-16. The VC with the highest priority (lowest priority level – 0) is allowed to send the next data frame as soon as it is ready. Within any level there can be any number of VCs which compete amongst themselves based on their bandwidth credit. A higher priority VC will always have precedence over a lower priority VC unless its Bandwidth Credit has reached the minimum credit limit in which case it is no longer allowed to send any more data frames. This prevents a high priority VC from consuming all the link bandwidth if it fails and starts babbling. More than one VC can be set to the same priority level in which case those VC's will compete for medium access using Bandwidth Reservation QoS [16, 17, 18].

5) Scheduled QoS: To provide fully deterministic data delivery it is necessary for the QoS mechanism to ensure that data from specific virtual channels can be sent (and delivered) at particular times. This can be done by chopping time into a series of time-slots (the schedule), during which a particular VC is permitted to send data frames. Each VC is allocated one or more time-slots in which it is permitted to send data frames. During a time-slot, if the VC is scheduled to send in that time-slot, it will compete with other VCs also scheduled to send in that time-slot based on precedence [16], [17], [18].

TABLE II. EXAMPLE OF THE SCHEDULE AND PRIORITY LEVELS

TIME SLOTS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
VC1 (0 - high priority)	0				0			
VC2 (0 - high priority)		0				0		
VC3			1	1		1	1	1
VC4			2	2			2	2
VC5			3	3			3	3
VC6 (4 - low priority)	4	4	4	4	4		4	4

B. *Quantity Calculation of Transmitted Streaming Data via SpaceFibre Output Virtual Channels*

Approach to quantity calculation of transmitted data via output VCs is divided into several steps which are described below in this subsection.

1) Schedule analyzing and subschedules defining: In this paper, a "sub-schedule" is a part of the schedule during which the values of the configuration parameters (that determine whether the virtual channel is assigned to the time slot, priority) are permanent, not changed. If such parts are available, then they form a sub-schedule for VCs. If there are no such parts, then all schedule is further considered as a common sub-schedule for all VCs.

Example: there are some 5 VCs. The priorities and time-slots (schedule table) are presented in Table III. As a result of the current step, 5 sub-schedules will be defined. They are:

- Sub-schedule 1: 0 – 63 time-slots;
- Sub-schedule 1: only 64 time-slot;
- Sub-schedule 3: 65 – 127 time-slots;
- Sub-schedule 4: only 128 time-slot;
- Sub-schedule 5: 129 – 190 time-slots;
- Sub-schedule 6: 191 – 255 time-slots.

TABLE III. DEFINING SUB-SCHEDULES

TIME SLOTS	**0**	**…**	**63**	**64**	**65**	**…**	**127**	**128**	**129**	**…**	**190**	**191**	**…**	**255**
VC1				1				1				1	1	1
VC2	0	0	0		0	0	0		0	0	0	0	0	0
VC3												4	4	4
VC4	2	2	2	3	2	2	2		2	2	2	3	3	3
VC5	1	1	1		1	1	1	2	1	1	1			
Sub-schedules	1			2	3			4	5			6		

2) Calculating of transmitted data: For each sub-schedule the average amount of transmitted data over the SpaceFibre output VCs is calculated. For this purpose the mathematical model of the streaming data transmission over output VCs for sub-schedule was developed. It is presented in the following subsections.

3) Summation: The total value of the average amount of transmitted data over output VCs is defined as the sum of all volumes calculated for each sub-schedules.

C. Math Model of Streaming over Output Virtual Channels

The process of streaming data transmission over output VCs could be represented as a queuing system (system, QS). The QS receives incoming requests (arrivals, calls). They form *N* streams with intensities: $\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda_N$ (Fig. 2). Each request is a data unit. The size of data unit does not exceed 256 bytes. The group of data units within a single stream forms a SpaceFibre packet. Each requests' stream corresponds to only one class of requests. Each requests class corresponds to only one VC. The total number of VCs is up to 256 VCs. In fact, only 32 VCs are used, the rest – reserved. Requests of different classes are buffered into different queues. All queues have the same limited capacity. The queue size varies from 1 to K requests. Each queue looks like a FIFO buffer of VC.

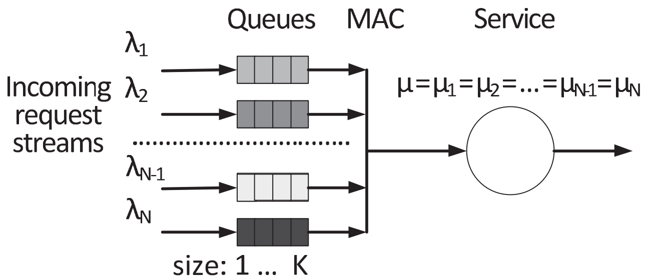

Fig. 2. Queue system of Output Virtual Channels SpaceFibre with MAC

The request service is the process of transmitting a data unit read from queue and then encapsulated into a data frame. The duration of data frame transmission does not depend on the request class. It is the same for all request classes. Requests are serviced with μ intensity.

The following assumptions are introduced:

- Data Link Layer of the SpaceFibre standard in the proposed mathematical model is considered.
- only Output Virtual Channels and Medium Access Controller mechanisms of Data Link Layer are considered.
- VCs are always notified that there is a room for a data frame at the other end of the link.
- It is not considered the case when Bandwidth Credit reaches values: ± Bandwidth Credit Limit or Minimum Bandwidth Credit Threshold.
- The process of writing/reading data into/from VC buffer and generating data frame is not considered, delays are not considered.
- Incoming requests' streams (bytes of SpaceFibre packets) are determined by Poisson process;
- Service times have an exponential distribution.

The intensity of incoming stream λ_i is calculated by the following formula:

$$\lambda_i = \frac{Size}{256 \cdot Period}\,, \tag{1}$$

where λ_1 – the intensity of incoming requests' stream; *Size* – size of SpaceFibre packet, byte; *256* is the constant of max.size of a data frame; *Period* – the average period (cycle) of receipt of SpaceFibre packet, μs.

The service time is defined by the following formula:

$$\mu = \frac{1}{t}, \quad t = \frac{S}{V}\,, \tag{2}$$

where μ – the service intensity; t – the duration of the service time, μs; S – the size of SpaceFibre data frame, bits; V – speed of SpaceFibre link, bit/μs.

There are priorities between the different request classes. It means that a request with higher priority is served before a request with lower priority. Once the request service is started, it cannot be disrupted until the whole service requirement is completed. Thus, only at the end of each service time one of the awaiting requests of the highest priority class is selected for the next service. The QS described above is a Markov random process with continuous time (continuous Markov chain). The states of the Markov chain (MC) could be represented as:

$$X = I \cup S, \tag{3}$$

where the disjoint sets I and S describe the states of the system:

- I – system is empty, all queues are empty;
- S – system busy states. System services a request of i class, $1 \le i \le$ N, where i - the number of the request class, $1 \le N \le 32$.

The state $S = \{S_1, S_2, \ldots, S_{N-1}, S_N\}$ is a finite set of system busy states:

- S_1 – it services a request of the 1st request class;
- S_2 – it services a request of the 2nd request class;

…

- S_{N-1} – it services a request of the N-1th request class;
- S_N – it services a request of the Nth request class.

Each state S_i is an enlarged state, because each busy state S_i is divided into own set of queue states: $V = \{V_0, V_1, \ldots, V_L\}$ $V \subseteq S_i$. For easy reading all states of V_k $(0 \le k \le L)$, they do not contain phrase "it services a request of i request class", but a reader have to bear it in mind:

- V_0: 0/0/…/0/0 – all queues are empty;
- V_1: 1/0/…/0/0 – one request is waiting for the service in the queue of the 1st request class, other queues are empty;
- V_1: 2/0/…/0/0 – two requests are waiting for the service in the queue of the 1st request class, other queues are empty;

…

- V_K: K/0/…/0/0 – queue of the 1st request class is full, other queues are empty;
- V_{K+1}: K/1/…/0/0 – queue of the 1st request class is full, one request is waiting for the service in the queue of the 2nd request class, other queues are empty;
- V_{K+2}: K/2/…/0/0 – queue of the 1st request class is full, two requests are waiting for the service in the queue of the 2nd request class, other queues are empty;

  …
- V_{K+K}: K/K/…/0/0 – queues of the 1st and 2nd request classes are full, other queues are empty;

  …
- $V_{K+K+...+1}$: K/K/…/1/0 – queues of the 1st, 2nd, …, N-2th request classes are full, one request is waiting for the service in the queue of the N-1th request class, another queue is empty;
- $V_{K+K+...+1}$: K/K/…/2/0 – queues of the 1st, 2nd, …, N-2th request classes are full, two requests are waiting for the service in the queue of the N-1th request class, another queue is empty;

  …
- $V_{K+K+...+K}$: K/K/…/K/0 – queues of the 1st, 2nd, …, N-2th, N-1th request classes are full, another queue is empty;
- $V_{K+K+...+K+1}$: K/K/…/K/1 – queues of the 1st, 2nd, …, N-2th, N-1th request classes are full, one request is waiting for the service in the queue of the Nth request class;
- $V_{K+K+...+K+1}$: K/K/…/K/2 – queues of the 1st, 2nd, …, N-2th, N-1th request classes are full, two requests are waiting for the service in the queue of the Nth request class;

  …
- $V_{L=K+K+...+K+K}$: K/K/…/K/K – all queues are full.

The state diagram of the system without queues' details is presented in Fig. 3.

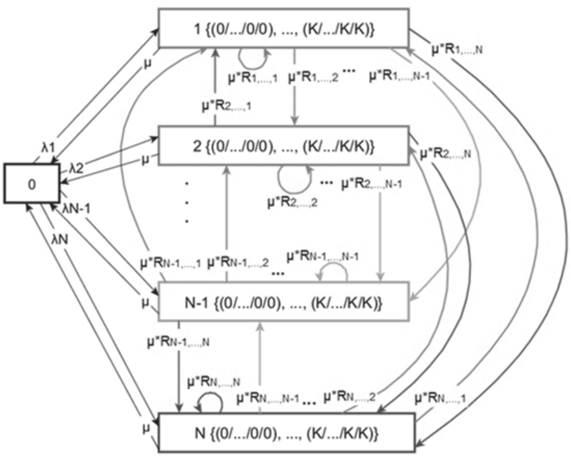

Fig. 3. The state diagram of the system without queues' details

The state diagram of the enlarged state S_i (i = 1) and queues' details is shown in

Fig. 4. The state diagram of the whole system is presented in Fig. 5.

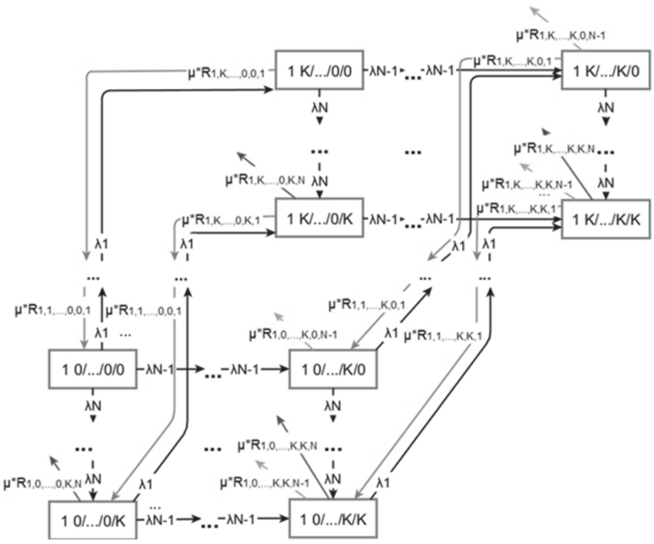

Fig. 4. The state diagram of the enlarged state S_i (i = 1) with queues' details

The distinctive feature of the proposed mathematical model is the set of probabilities $R_{i_0,i_1,...i_{N-1},i_N,i_{N+1}}$ ($i_a = 0,1,2,...,N$; $a = 1,2,...,N+1$). It is specified for each S_i with respect to V_k queue state. The following index notation is used: i_0 – it specifies a request class's number of served request: $1 \le i_0 \le N$; $i_1,...,i_{N-1},i_N$ – indexes that define one particular state of the queue states of V_k: $i_b = 0,1,2,...,K; b = 1,2,...,N$; i_{N+1} – it specifies index of a probability to make the transition of other states: $1 \le i_{N+1} \le N$.

$$\sum_{j=1}^{N} R_{i,0,...,0,0,j} = 1, \quad (4)$$

$$\sum_{j=1}^{N} R_{i,0,...,0,0,j} + ... + \sum_{j=1}^{N} R_{i,K,...,K,K,j} = W_i = L+1, \quad (5)$$

$$\sum_{i=1}^{N} W_i + 1 = (L+1) \cdot N + 1 = Q, \quad (6)$$

where Q is the total number of all system states.

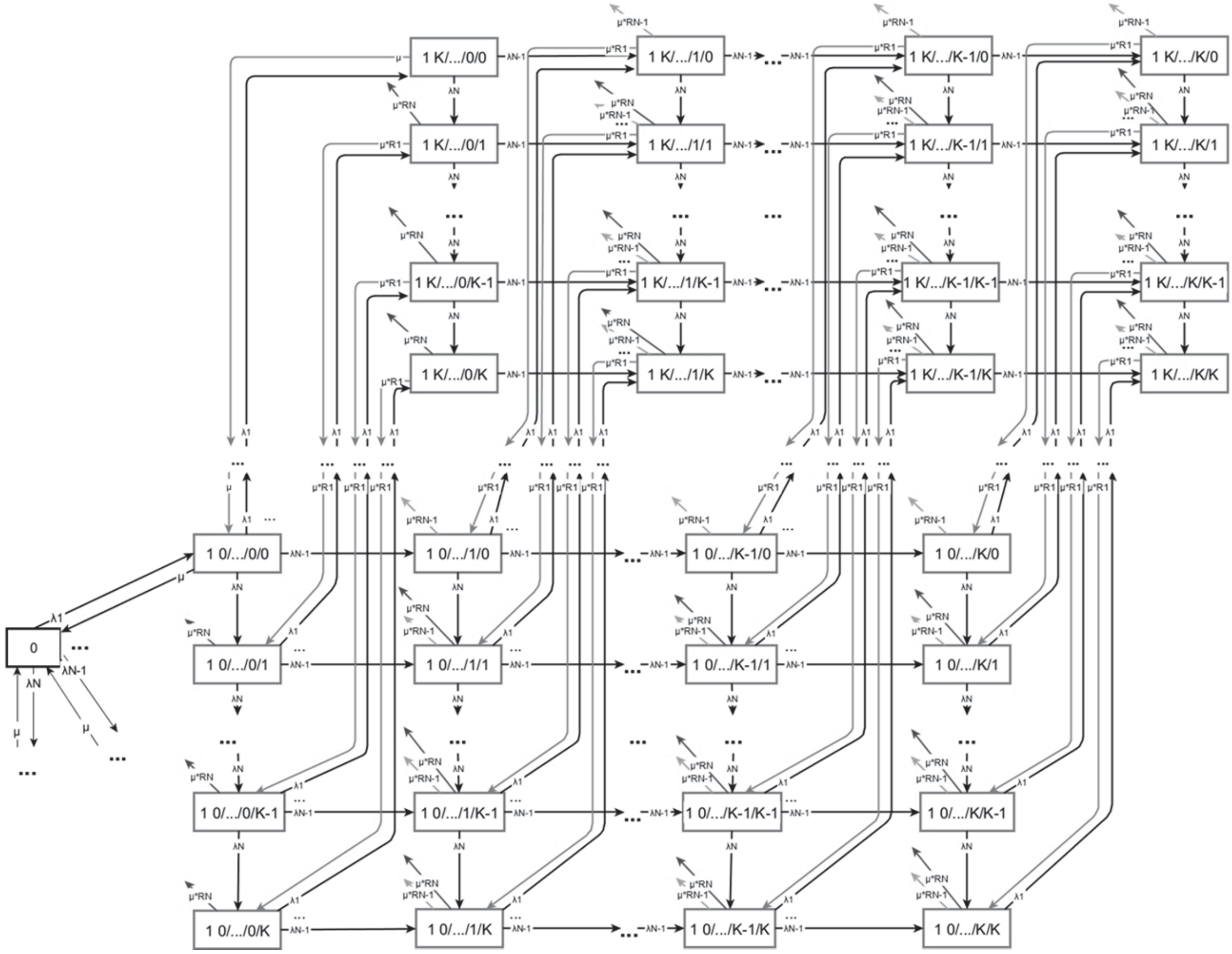

Fig. 5. The state diagram of the enlarged state S_i (i = 1) with queues' details (for ease of reading $R_{i_0,\ldots,i_{N+1}}$ is reduced to $R_{i_{N+1}}$)

The probabilities $R_{i_0,\ldots,i_{N+1}}$ allow to make the transition to service of a request from the queue of i_b request class. Such kind of transition will move the system to one state of queue states V_k $(0 \le k \le L)$ within the current enlarged state S_j $(j = i_b)$. It also permits to move the system from the current enlarged state S_j to S_{i_b} $(j \ne i_b)$. As a result, the system will be transferred to one of the states of queue state V_k for the enlarged state S_{i_b}. The function F is introduced to determine the probabilities $R_{i_0,\ldots,i_{N+1}}$:

$$R_{i_0,\ldots,i_{N+1}} = F(M, E, C, V)\,, \qquad (7)$$

where M – priority levels of all VCs: $M = \{M_1, M_2, \ldots, M_N\}$ $0 \le M_i \le 16$; E – percentage of overall link bandwidth that VC is expected to use: $E = \{E_1, E_2, \ldots, E_N\}$ $\sum_{i=1}^{N} E_i \le 0.9$; C – maximum amount of link bandwidth that VC is allowed to accumulate; V – numbers of queues that contain requests for service.

Using the state diagram shown in Fig. 5 the system of differential Kolmogorov's equations could be obtained. It is necessary to point out that the following situation is considered [18]:

- there are the steady-state conditions;
- there are steady-state probabilities of the system states.

It could be transformed to the system of linear equations (SLE). The solution of SLE is the steady-state probabilities $p_{i_0,i_1,\ldots,i_{N-1},i_N}$ (index notation just like $R_{i_0,i_1,\ldots i_{N-1},i_N,i_{N+1}}$). These probabilities mean the average relative time of request service time. This time determines the whole duration of request transmitting of i_0 request class. The SLE equations are presented:

$$\begin{cases}
\sum_{i=1}^{N}\lambda_i p_{0,0,\dots,0,0} = \mu\sum_{i=1}^{N} p_{1,0,\dots,0,0}, \\
\left(\sum_{i=1}^{N}\lambda_i + \mu\right)p_{1,0,\dots,0,0} = \lambda_1 p_{0,0,\dots,0,0} + \mu\sum_{i=1}^{N} p_{i,1,\dots,0,0}, \\
\left(\sum_{i=1}^{N}\lambda_i + \mu\right)p_{1,q,\dots,0,0} = \lambda_1 p_{1,q-1,\dots,0,0} + \mu\sum_{i=1}^{N} p_{i,q+1,\dots,0,0}, q=\overline{1,K-1}, \\
\left(\sum_{i=2}^{N}\lambda_i + \mu\right)p_{1,K,\dots,0,0} = \lambda_1 p_{1,K-1,\dots,0,0}, \\
\dots \\
\left(\sum_{i=N-1}^{N}\lambda_i + \mu\sum_{j=1}^{N-1} R_{1,K,\dots q,0,j}\right)p_{1,K,\dots,q,0} = \lambda_1 p_{1,K-1,\dots,q,0} + \dots + \lambda_{N-1} p_{1,K,\dots,q-1,0}, q=\overline{1,K-1}, \\
\left(\lambda_N + \mu\sum_{j=1}^{N-1} R_{1,K,\dots K,0,j}\right)p_{1,K,\dots,K,0} = \lambda_1 p_{1,K-1,\dots,K,0} + \dots + \lambda_{N-1} p_{1,K,\dots,K-1,0}, \\
\dots \\
\left(\lambda_N + \mu\sum_{j=1}^{N} R_{1,K,\dots K,q,j}\right)p_{1,K,\dots,K,q} = \lambda_1 p_{1,K-1,\dots,K,q} + \dots + \lambda_N p_{1,K,\dots,K,q-1}, q=\overline{1,K-1}, \\
\left(\mu\sum_{j=1}^{N} R_{1,K,\dots K,K,j}\right)p_{1,K,\dots,K,K} = \lambda_1 p_{1,K-1,\dots,K,K} + \dots + \lambda_N p_{1,K,\dots,K,K-1}, \\
\dots \\
\left(\mu\sum_{j=1}^{N} R_{N,K,\dots K,K,j}\right)p_{N,K,\dots,K,K} = \lambda_1 p_{N,K-1,\dots,K,K} + \dots + \lambda_N p_{N,K,\dots,K,K-1}
\end{cases} \tag{8}$$

$$p_{0,0,\dots,0,0} + \sum_{j=0}^{K} p_{1,0,\dots,0,j} + \sum_{j=0}^{K} p_{1,0,\dots,1,j} + \dots + \sum_{j=0}^{K} p_{N,K,\dots,K,j} = 1$$

From knowledge of the average duration of the service time and size of request in bytes the average quantity of transmitted data of i_0 request class could be calculated by the following formula:

$$AQ_{i_0} = \frac{T_{i_0}\cdot Length_{i_0}}{T_{frame}\cdot Frames_{i_0}}, \tag{9}$$

where AQ_{i0} – the average quantity of transmitted data of i_0 request class, bytes; T_{i0} – the average duration of the requests' service time of i_0 request class, ms; T_{frame} – the average time for sending a data frame, ms; $Length_{i0}$ – the length of packet of i_0 request class, bytes; $Frames_{i0}$ – the quantity of frames inside one packet of i_0 request class.

AQ_{i0} will demonstrate the average amount of transmitted data over output VCs, because request classes are brought in line with VCs.

D. *Example of Calculations*

There are considered 2 VCs, sizes of VCs buffers are up to 1 request. Traffic parameters are demonstrated in Table IV. Parameters of VCs are presented in Table V.

TABLE IV. TRAFFICS PARAMETERS

Request classes	Size, byte	Quantity frames in packet	Average period of receipt, us
VC1	1024	4	20
VC2	260	1,015625	5000

TABLE V. VCS PARAMETERS

Request classes	Priority level	Expected Bandwidth	0	...	15	16	17	…	31	32	33	...	46	47	…	63
VC1	1	0,032768				1				1				1	1	1
VC2	0	0,0003328	0	0	0		0	0	0		0	0	0	0	0	0
Duration of 1 time-slot, us			260,9375													
SpaceFibre link, Gbit/sec			1,25													

There is only one sub-schedule: 0 – 63 time-slots. The intensities are: λ_1 = 0,00020312; λ_2 = 0,2; μ = 0,473485. The state diagram is shown in Fig. 6.

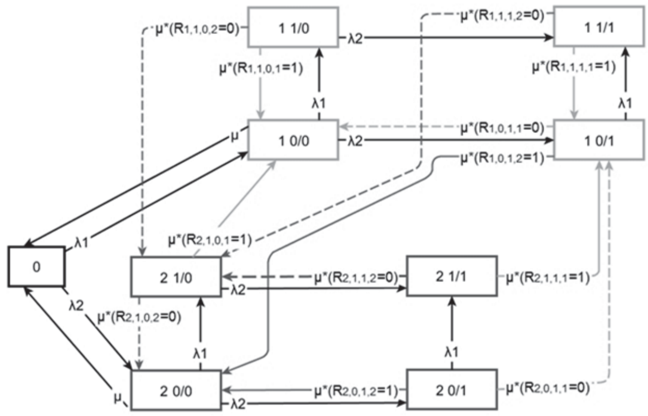

Fig. 6. The state diagram of the system with 2 VCs and buffer for 1 request

The SLE equations are presented below:

$$\begin{cases}
(\lambda_1+\lambda_2)p_{0,0,0} = \mu(p_{1,0,0}+p_{2,0,0}), \\
(\mu+\lambda_1+\lambda_2)p_{1,0,0} = \lambda_1 p_{0,0,0}+\mu(p_{1,1,0}+p_{2,1,0}), \\
(\mu+\lambda_2)p_{1,1,0} = \lambda_1 p_{1,0,0}, \\
(\mu+\lambda_1)p_{1,0,1} = \lambda_2 p_{1,0,0} + \mu(p_{1,1,1}+p_{2,1,1}), \\
\mu p_{1,1,1} = \lambda_2 p_{1,1,0} + \lambda_1 p_{1,0,1}, \\
(\mu+\lambda_1+\lambda_2)p_{2,0,0} = \lambda_2 p_{0,0,0} + \mu(p_{1,0,1}+p_{2,0,1}), \\
(\mu+\lambda_1)p_{2,0,1} = \lambda_2 p_{2,0,0}, \\
(\mu+\lambda_2)p_{2,1,0} = \lambda_1 p_{2,0,0}, \\
\mu p_{2,1,1} = \lambda_1 p_{2,0,1} + \lambda_2 p_{2,1,0}.
\end{cases} \tag{10}$$

$$p_{0,0,0} + \sum_{j=0}^{K=1} p_{1,0,j} + \sum_{j=0}^{K=1} p_{1,1,j} + \sum_{j=0}^{K=1} p_{2,0,j} + \sum_{j=0}^{K=1} p_{2,1,j} = 1$$

The solution of the equations is shown in formula (11).

After substituting numbers, the steady-state probabilities of all system states are: $p_{0,0,0}$ = 0.624319541; $p_{1,0,0}$ = 0.000244197; $p_{1,1,0}$ = 0.000000074; $p_{1,0,1}$ = 0.00018455; $p_{1,1,1}$ = 0.00000011; $p_{2,0,0}$ = 0.26373621; $p_{2,0,1}$ = 0.111354404; $p_{2,1,0}$ = 0.000079544; $p_{2,1,1}$ = 0.00008137.

If the running time of SpaceFibre switch is 500 ms, the average duration of the service time of VC1 requests (taking into account time-slots) will be 55.7 ms; the average service time of VC2 requests is 0.21 ms.

The quantity of transmitted data over VCs: AQ_{VC1} = 6.4 MBytes; AQ_{VC2} = 24,6 KBytes. It could be converted to VCs throughputs: 11.8 KBytes/ms and 0.05 KBytes/ms.

$$\begin{cases}
p_{0,0,0} = \dfrac{\mu^3*(\mu+\lambda 2)}{\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2}, \\
p_{1,0,0} = \dfrac{\mu^3*\lambda 1*(\lambda 1+2*\lambda 2+\mu)}{\left(\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2\right)*(\mu+\lambda 1+\lambda 2)}, \\
p_{1,1,0} = \dfrac{\mu^3*\lambda 1^2*(\lambda 1+2*\lambda 2+\mu)}{\left(\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2\right)*(\mu+\lambda 2)*(\mu+\lambda 1+\lambda 2)}, \\
p_{1,0,1} = \dfrac{\mu*\lambda 2*\left(\mu^3+(3*\lambda 1+4*\lambda 2)*\mu^2+3*(\lambda 1+\lambda 2)^2*\mu+(\lambda 1+\lambda 2)^3\right)*\lambda 1}{\left(\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2\right)*(\mu+\lambda 1)*(\mu+\lambda 1+\lambda 2)}, \\
p_{1,1,1} = \dfrac{\left(2*\left(\left(\frac{1}{2}\right)*\lambda 2+\mu+\left(\frac{1}{2}\right)*\lambda 1\right)\right)*\lambda 2*\lambda 1^2*\left(\lambda 2^3+(2*\lambda 1+2*\mu)*\lambda 2^2+\left(\lambda 1^2+3*\lambda 1*\mu+3*\mu^2\right)*\lambda 2+\mu*(\mu+\lambda 1)^2\right)}{\left((\mu+\lambda 1)*\lambda 2^3+2*(\mu+\lambda 1)^2*\lambda 2^2+\left(\lambda 1^3+3*\lambda 1^2*\mu+3*\lambda 1*\mu^2+2*\mu^3\right)*\lambda 2+\mu^2*\left(\lambda 1^2+\lambda 1*\mu+\mu^2\right)\right)*(\mu+\lambda 2)*(\mu+\lambda 1)*(\mu+\lambda 1+\lambda 2)}, \\
p_{2,0,0} = \dfrac{\mu^2*\lambda 2*\left(\lambda 2^2+(2*\lambda 1+2*\mu)*\lambda 2+\mu^2+\lambda 1*\mu+\lambda 1^2\right)}{\left((\mu+\lambda 1)*\lambda 2^3+2*(\mu+\lambda 1)^2*\lambda 2^2+\left(\lambda 1^3+3*\lambda 1^2*\mu+3*\lambda 1*\mu^2+2*\mu^3\right)*\lambda 2+\mu^2*\left(\lambda 1^2+\lambda 1*\mu+\mu^2\right)\right)*(\mu+\lambda 1+\lambda 2)}, \\
p_{2,0,1} = \dfrac{\mu^2*\lambda 2^2*\left(\mu^2+(\lambda 1+2*\lambda 2)*\mu+(\lambda 1+\lambda 2)^2\right)}{\left(\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2\right)*(\mu+\lambda 1)*(\mu+\lambda 1+\lambda 2)}, \\
p_{2,1,0} = \dfrac{\mu^2*\lambda 2*\left(\lambda 2^2+(2*\lambda 1+2*\mu)*\lambda 2+\mu^2+\lambda 1*\mu+\lambda 1^2\right)*\lambda 1}{\left((\mu+\lambda 1)*\lambda 2^3+2*(\mu+\lambda 1)^2*\lambda 2^2+\left(\lambda 1^3+3*\lambda 1^2*\mu+3*\lambda 1*\mu^2+2*\mu^3\right)*\lambda 2+\mu^2*\left(\lambda 1^2+\lambda 1*\mu+\mu^2\right)\right)*(\mu+\lambda 2)*(\mu+\lambda 1+\lambda 2)}, \\
p_{2,1,1} = \dfrac{\left(2*\left(\left(\frac{1}{2}\right)*\lambda 2+\mu+\left(\frac{1}{2}\right)*\lambda 1\right)\right)*\mu*\lambda 2^2*\left(\mu^2+(\lambda 1+2*\lambda 2)*\mu+(\lambda 1+\lambda 2)^2\right)*\lambda 1}{\left(\mu^4+(\lambda 1+2*\lambda 2)*\mu^3+(\lambda 1+2*\lambda 2)*(\lambda 1+\lambda 2)*\mu^2+(3*(\lambda 1+\lambda 2))*\lambda 2*\left(\lambda 1+\left(\frac{1}{3}\right)*\lambda 2\right)*\mu+\lambda 1*\lambda 2*(\lambda 1+\lambda 2)^2\right)*(\mu+\lambda 2)*(\mu+\lambda 1)*(\mu+\lambda 1+\lambda 2)}.
\end{cases} \quad (11)$$

III. Conclusion

The paper gave a short overview of actual problem of streaming data transfer over onboard spacecraft networks. The SpaceFibre standard could be used for streaming. The mathematical approach to quantity calculation of transmitted streaming data via SpaceFibre was developed to evaluate how much data will be transferred via SpaceFibre. The mathematical model of the streaming data transfer over SpaceFibre output virtual channels was also proposed. All these results are actual and significant for designers of spacecraft networks. It would make the process of evaluation of the SpaceFibre network characteristics and generating network configuration parameters (priorities, time-slots, etc.) much easier.

Acknowledgment

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation within the project part of the federal assignment under contract N 8.4048.2017/4.6 from 31.05.2017.

References

[1] E. Mikrin, *Design concept of Spacecraft Onboard Control System*. Moscow: Bauman MSTU Publ., 2003, 336 p.

[2] I. Korobkov, "Adaptive Control of the Network Protocols' Mechanisms in the Onboard Networks", *Proceedings of SUAI Scientific session*, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, SUAI, 2015, pp. 73-82.

[3] Y. Sheynin, E. Suvorova, I. Korobkov, J. Petrichkovich, T. Solokhina, A. Glushkov, A. Sakharov, "Intelligent Networking for Distributed High-Rate Streaming Sensor Fields," *in Proceedings of 16th edition of Sophia Antipolis MicroElectronics (SAME) Forum 2013,* Sophia Antipolis, 2013.

[4] I. Korobkov, E. Suvorova, Y. Sheynin, V. Olenev "Streaming Services over SpaceFibre Networks", *Proceedings of 7th International SpaceWire Conference 2016 (ISC2016)*, Yokohama, Japan, 2016. pp. 151-158.

[5] I. Korobkov "Adaptive Data Streaming Service for Onboard Spacecraft Networks", *Proceedings of 17th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT)*, P.G. Demidov Yaroslavl State University, Yaroslavl, Russia, 2015, pp. 291-298.

[6] A. Khakhulin, I. Orlovsky, Y. Sheynin, I. Korobkov, V. Olenev, E. Suvorova, I.Lavrovskaya "SpaceFibre Based On-board Networks for Real-Time Video Data Streams", *Proceedings of 7th International SpaceWire Conference 2016 (ISC2016)*, Yokohama, Japan, 2016, pp. 327-333.

[7] N. Matveeva, E. Suvorova, Y. Sheynin, "QoS mechanisms in SpaceFibre and RapidIO Services over SpaceFibre Networks", *Proceedings of 7th International SpaceWire Conference 2016 (ISC2016)*, Yokohama, Japan, 2016, pp. 305-312.

[8] N. Matveeva, E. Suvorova "Different Implementation of Network Level in Embedded Networking with QoS", *Proceedings of 14th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT)*, Helsinki, Finland, 2013, pp. 67-76.

[9] N. Matveeva, E. Suvorova, V. Olenev, I. Lavrovskaya, I. Korobkov, Artur Eganyan "SpaceFibre Quality of Service Features Support in the Network Level", *Proceedings of 5th International Conference SpaceWire 2013 (ISC2013)*, Gothenburg, 2013, pp. 158-165.

[10] N. Matveeva, E. Suvorova "Network Layer Support in SpaceFibre Protocol", *Proceedings of 6th International Conference SpaceWire 2014 (ISC2014)*, Athens, Greece, 2014, pp. 224-228.

[11] N. Matveeva, Y. Sheynin, E. Suvorova "QoS in SpaceFibre and SpaceWire/GigaSpaceWire Protocols", *Proceedings of 6th International Conference SpaceWire 2014 (ISC2014)*, Athens, Greece, 2014, pp. 145-152.

[12] Y. Sheynin, E. Suvorova, N. Matveeva, A. Khakhulin, I. Orlovsky, D. Romanov "SpaceFibre Based Spacecraft Network Case Study", *Proceedings of 6th International Conference SpaceWire 2014 (ISC2014)*, Athens, Greece, 2014, pp. 164-168.

[13] A. Shamshin, I. Korobkov, N. Matveeva "Evaluations of the SpaceFibre QoS Mechanisms for Streaming", *Proceedings of SUAI Scientific session*, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, SUAI, 2015, pp. 178-191.

[14] E. Suvorova, N. Matveeva, I. Korobkov, A. Shamshin, Y. Sheynin, "Virtual Prototyping in SpaceFibre System-on-Chip Design", *Design and Verification Conference and Exhibition (DvCon) Europe 2015*, Munich, Germany, 2015.

[15] I. Korobkov, E. Suvorova, N. Matveeva, Y. Sheynin, "Virtual Prototyping for Spacefibre Standard Research", *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*, volume 18, No.1(2), Samara, Russia, 2016, pp. 390-396.

[16] V. Olenev, I. Lavrovskaya, I. Korobkov, "SpaceWire-RT/SpaceFibre Specification and Modeling", *Proceedings of 5th International Conference SpaceWire 2013 (ISC2013)*, Gothenburg, 2013, pp. 344-350.

[17] S. Parkes, A. Ferrer, A. Gonzalez, C. McClements, *SpaceFibre Specification*. Draft J1, 2017, 161 p.

[18] S. Parkes, C. McClements, D. McLaren, A.Ferrer, A.Gonzalez, "SpaceFibre: Multi-Gigabit/s Interconnect for Spacecraft On-board Data Handling", *IEEE Aerospace Conference*, Big Sky, Montana, 2015.

[19] E. Ventsel, *Operations Research.* Moscow: Soviet Radio, 1972, 552 p.