

# Different Implementation of Network Level in Embedded Networking with QoS

Nadezhda Matveeva, Elena Suvorova

Saint-Petersburg State University of Aerospace Instrumentation  
Saint-Petersburg, Russian Federation  
n.matveeva88@gmail.com, suvorova@aanet.ru

**Abstract**—Some modern standards in space industry, which are being used in embedded networking designs, provide quality of service features, which are implemented by means of virtual channels. Implementations of virtual channels mechanisms are very different. Each implementation has its latency characteristics for packet flow, hardware cost and performance. These parameters depend on the virtual channels quantity in a port and switch matrix's channels quantity connected to every port (connection point). The connection point quantity can vary from one to a number of virtual channels in port. We consider three structures and implementations of network layer. In the first implementation quantity of connection points is equal to number of virtual channels in a port. In the second - one connection point. The third – one connection point with lower priority data transmission interruption. In this article we compare characteristics of different architecture implementations and structures of port controllers and switch matrix. Also we analyze and simulate proposed mechanism. We present formulas to calculate minimum and maximum data packet transmission latency and compare theoretic and simulation results. Count of virtual channels is 4 for simulation, packet length – 250, 750 bytes. Moreover router's switch matrix hardware cost is evaluated in the article.

**Keywords**—Embedded networking, SpaceFibre, Virtual channel.

## I. INTRODUCTION

Performance of modern embedded systems depends on network architecture and structure. Existing embedded networks support data transmission with Quality of Service (QoS) [1]. Currently many different standards are widely used in design of network. For example – RapidIO [2], SpaceWire [3] and etc. It supports data transmission on virtual channels.

For our research we chose different approaches to implementation technology of virtual channels [4]. The first allows transferring data at the same time from different virtual channels of a port. The second – only one virtual channel of a port can transfer data. The third - virtual channel with higher priority can interrupt the transmission of data with lower priority. These approaches are not associated with a specific standard. It can be used in the

construction of different embedded networking technologies [5].

We will use SpaceFibre in our case study. SpaceFibre is the modern standard in space industry. This technology also can be used for construction embedded networks.

SpaceFibre provides a coherent quality of service (QoS) mechanism able to support best effort, bandwidth reserved, scheduled and priority based qualities of service.[6] Quality of service parameters [7] that can be provided by routers with SpaceFibre ports depend not only on the SpaceFibre protocol characteristics and port specific implementation but also on a network layer implementation. In this article we analyze different implementation of network layer SpaceFibre.

## II. STRUCTURES OF NETWORK LEVEL

### A. 1<sup>st</sup> way of router's network layer structure

Router's switch matrix includes a separate channel for connection of each input virtual channel with the correspondent output virtual channel in this way. Quantity of connection points to the switch matrix (hereinafter – connection points) for every port of a router is equal to the virtual channels number in this port, Fig. 1 (only one data transmission direction is represented). This way was recommended by the SpaceWire-RT specification draft [8]. In such router structure data flows can compete with each other only within one virtual channel in output port of router. In this case timing characteristics in the network layer depend only on arbitration rules. In all other cases timing characteristics of data flows are not influenced by the router network layer. However, such router structure results in an essential hardware cost.

### B. 2<sup>nd</sup> way of router's network layer structure

According to this router structure, the quantity of connection points for every port is less than number of virtual channels in the port. There is one connection point. In our research we suppose that data flows from every virtual channel can be transmitted via one connection point of the correspondent port, Fig. 2. Hardware cost of this router structure is essentially less, than hardware cost of the

previous one. But in this way, data flows from different virtual channels share switch matrix channels. Therefore, an impact between data flows and corresponding disturbance of its timing characteristics in this case in this router structure is more essential than in the previous one.

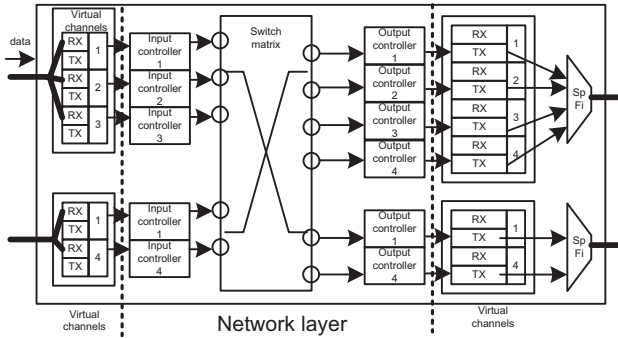


Fig. 1 The first way of router's network layer implementation

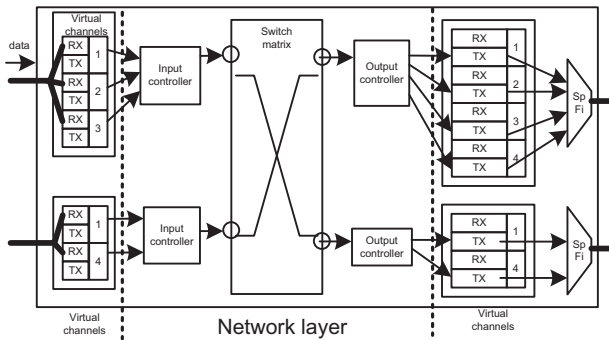


Fig. 2 The second way of router's network layer implementation

### C. 3<sup>rd</sup> way of router's network layer structure

This router structure is similar to 2<sup>nd</sup> way. The difference between these ways is possibility of lower priority data transmission interruption. Condition of data transmission interruption can be different. Packet transmission can be interrupted after N byte transfer.

## III. THEORETICAL PARAMETERS EVALUATION

Maximum/minimum delays are calculated for the proposed 2<sup>nd</sup> way router structure. The following assumptions were made during calculations: the packet size for the virtual channel was the same for every source; for every virtual channel data transmission is enabled in every time slot; Nchars are written to TX and RX buffers of each port at the same amount of time; the packet size for every virtual channel is less then frame size; the frame size is less then buffer size for every virtual channel; for every port of a device has the same value.

Notation:

$i$  - an identifier of a virtual channel;

$k$  - an identifier of a node (a terminal node or a router);  
 $l$  - an identifier of a link;  
 $p$  - an identifier of a port ;  
 $h$  - an identifier of a virtual channel with the highest priority;

$sizeF$  - the frame size in bytes;

$sizeP\_VC_i$  - a packet size for the virtual channel  $i$  in bytes;

$sizeB\_VC_i$  - a buffer size for the virtual channel  $i$  in bytes;

$countSw\_VC_i$  - the number of routers which should be passed for transmission of data of the virtual channel  $i$ ;

$countLink\_VC_i$  - the number of links which should be passed for transmission of data of the virtual channel  $i$ .

$$countLink\_VC_i = countSw\_VC_i + 1$$

$v_l$  - a data rate in the link  $l$ , Gb/s.

$Tbyte_l$  - the transmission time of 1 Nchar (1 byte) through the link  $l$ .

$$Tbyte_l = \frac{1}{v_l}$$

$f_k$  - an operating frequency of the node  $k$ , MHz.

$minDelay\_VC_i$  - the minimal packet's transmission delay for the virtual channel  $i$  for the whole transmission path;

$maxDelay\_VC_i$  - maximal packet's transmission delay for virtual channel  $i$  for the whole transmission path;

$\{SourceVC_i\}$  - a set of source nodes for the virtual channel  $i$ ;

$\{DestinVC_i\}$  - a set of destination nodes for the virtual channel  $i$ ;

$TwrByteTX_k$  - time of writing of 1 Nchar into the TX buffer of the node  $k$ ;

$TwrByteRX_k$  - time of writing of 1 Nchar into the RX buffer of the node  $k$  (for this implementation it is equal to the transmission time of 1 byte through the SpaceFibre link, i.e.  $TwrByteRX_k = Tbyte_l$ )

$TcalcPrec_k$  - time of the Precedence calculation for all virtual channels in node  $k$ . This parameter is defined by the developer of the system.

$DelaySwMatrix_k$  - the delay of accessing to routing table and selection of connection points in a router with identifier  $k$ . This time is necessary to connect the input port with the output port for data transmission.

A. Calculation of the minimum data transmission delay for the virtual channel  $i$

$\{minLinkVC_i\}$  - a set of physical links, which constitute the shortest data transmission path for the virtual channel  $i$ .

$\{minSwVC_i\}$  - a set of routers, which constitute the shortest data transmission path for the virtual channel  $i$ ;

$minDelaySource\_VC_i$  - the minimal processing delay in a packet's source of the virtual channel  $i$

$$\begin{aligned} minDelaySource\_VC_i &= \\ &= \min_{k \in \{SourceVC_i\}} (sizeP\_VC_i \cdot TwrByteTX_k \\ &\quad + TcalcPrec_k) \end{aligned}$$

$minDelayDestin\_VC_i$  - the minimal processing delay in a receiver of the virtual channel  $i$

$$\begin{aligned} minDelayDestin\_VC_i &= \\ &= \min_{k \in \{DestinVC_i\}} (sizeP\_VC_i \cdot TwrByteRX_k) \end{aligned}$$

$minDelaySw_k\_VC_i$  - the minimal delay in a router for packets of the virtual channel  $i$ . We assume that there is no competition between packets of one virtual channel and that different virtual channels do not compete in the router's output port.

$$\begin{aligned} minDelaySw_k\_VC_i &= \\ &= sizeP\_VC_i \cdot TwrByteRX_k \\ &\quad + DelaySwMatrix_k + TcalcPrec_k \\ &\quad + sizeP\_VC_i \cdot TwrByteTX_k \end{aligned}$$

$$\begin{aligned} minDelay\_VC_i &= minDelaySource\_VC_i \\ &\quad + \sum_{k \in \{minSwVC_i\}} minDelaySw_k\_VC_i \\ &\quad + minDelayDestin\_VC_i \end{aligned}$$

B. Calculation of the maximum data transmission delay for the virtual channel  $i$

$\{maxLinkVC_i\}$  - a set of links, which constitute the longest data transmission path for the virtual channel  $i$ ;

$\{maxSwVC_i\}$  - a set of routers, which constitute the longest data transmission path for virtual channel  $i$ ;

$maxDelaySource\_VC_i$  - the maximum processing delay in a packet's source of the virtual channel  $i$ ;

$\{allVC_p\}$  - a set of virtual channels, which are supported in the port with identifier  $p$  of a node.

$\{allVCHighPriority_i\}$  - a set of virtual channels, which are supported in the port with identifier  $p$  of a node and have a higher priority than the priority of the virtual channel  $i$ ;

$\{allPortVC_i\}$  - a set of node's ports which support data transmission via the virtual channel  $i$ ;

$$\begin{aligned} maxDelaySource\_VC_i &= \\ &= \max_{k \in \{SourceVC_i\}} \{ \max_{p \in \{allPortVC_i\}} (sizeP\_VC_i \\ &\quad \cdot TwrByteTX_k + TcalcPrec_k) \\ &\quad + \sum_{j \in \{allVC_p\}, j \neq i} (sizeP\_VC_j \cdot TwrByteTX_k + TcalcPrec_k) \} \end{aligned}$$

$maxDelayDestin\_VC_i$  - the maximum processing delay in a destination node for packets of the virtual channel  $i$

$$\begin{aligned} maxDelayDestin\_VC_i &= \\ &= \max_{k \in \{DestinVC_i\}} (sizeP\_VC_i \\ &\quad \cdot TwrByteRX_k) \end{aligned}$$

$maxDelaySw_k\_VC_i$  - maximum delay in a router for packets of the virtual channel  $i$  for the case when the competition exists between the packets of one virtual channel and the packets of different virtual channels for the switch output port.

$$\begin{aligned} maxDelaySw_k\_VC_i &= \\ &= sizeP\_VC_i \cdot TwrByteRX_k \\ &\quad + DelaySwMatrix_k + TcalcPrec_k \\ &\quad + (\{allPortVC_i\} - 1) \cdot (sizeP\_VC_i \\ &\quad \cdot TwrByteRX_k \\ &\quad + \sum_{j \in \{allVCHighPriority_i\}, j \neq i} (sizeP\_VC_j \\ &\quad \cdot TwrByteTX_k + TcalcPrec_k) \\ &\quad + TcalcPrec_k) \end{aligned}$$

$$\begin{aligned} maxDelay\_VC_i &= maxDelaySource\_VC_i \\ &\quad + \sum_{k \in \{maxSwVC_i\}} maxDelaySw_k\_VC_i \\ &\quad + maxDelayDestin\_VC_i \end{aligned}$$

C. Calculation of the maximum/minimum data transmission delay for the virtual channel with the highest priority

The following restrictions were made during calculations: for every virtual channel data transmission is enabled in every time slot; all routers contain only one connection point for each port.. The connection point is shared by all virtual channels of the corresponding port.

$minDelay\_VC_h$  - the minimal packet transmission delay for the virtual channel with the highest priority for the whole transmission path.

$maxDelay\_VC_h$  - the maximal packet transmission delay for the virtual channel with the highest priority for the whole transmission path.

The value of the minimal packet transmission delay for the virtual channel with the highest priority is equal to the value of the minimal packet transmission delay for the virtual channel of an arbitrary priority.

$$minDelay\_VC_h = minDelay\_VC_i$$

The value of the maximal packet transmission delay for the virtual channel with the highest priority is not equal to the minimal packet transmission delay for the virtual channel of an arbitrary priority.

$maxDelaySource\_VC_h$  - the maximal packet processing delay for the virtual channel with the highest priority in a source node.

$$\begin{aligned}
 & \maxDelaySource\_VC_h \\
 &= \max_{k \in \{SourceVC_h\}} \{ \max_{p \in \{allPortVC_h\}} (sizeP\_VC_h \\
 & \cdot TwrByteTX_k + (sizeF - 1) \cdot TwrByteTX_k \\
 & + TcalcPrec_k) \}
 \end{aligned}$$

$\maxDelayDestin\_VC_h = \maxDelayDestin\_VC_i$  - the maximal packet processing delay in a destination node for the highest priority virtual channel is equal to the maximal packet processing delay in a destination node of an arbitrary virtual channel priority.

$\maxDelaySw_k\_VC_h$  - the maximal delay in a router for packets from the virtual channel with the highest priority. We assume that the frame of the lower priority packet is already being transmitted and there is a competition between the packets of virtual channels of the same priority in output port of router.

$$\begin{aligned}
 & \maxDelaySw_k\_VC_h \\
 &= sizeP\_VC_h \cdot TwrByteRX_k \\
 &+ DelaySwMatrix_k + TcalcPrec_k \\
 &+ (sizeF - 1) \cdot TwrByteRX_k \\
 &+ (|\{allPortVC_h\}| - 1) \cdot (sizeP\_VC_h \\
 & \cdot TwrByteRX_k + TcalcPrec_k) \\
 & \maxDelay\_VC_h = \maxDelaySource\_VC_h \\
 &+ \sum_{k \in \{maxSwVC_h\}} \maxDelaySw_k\_VC_h \\
 &+ \maxDelayDestin\_VC_h
 \end{aligned}$$

#### IV. RESULTS OF TIMING CHARACTERISTICS

##### A. Network model

Timing characteristics estimation was done on the basis of the models, which are depicted in Fig. 3.

The Network model 1 comprises a router with 4 ports, each of which can work with 4 virtual channels. Terminal nodes generate packets in a random time moments. At these random moments the terminal node sends the generated packets to each virtual channel. The destination nodes for each virtual channel are also chosen randomly and can be different for the virtual channels. This configuration can lead to a potential possibility of data packets flow concurrency in the output port.

##### B. Simulation

The network was simulated on the adapted DCNSimulator model. DCNSimulator is based on Qt and SystemC. It consists of the simulation engine and libraries of network components. The simulation engine is the general part that could work for simulation of any network. Libraries of network components are specific for particular network standards and could represent network components

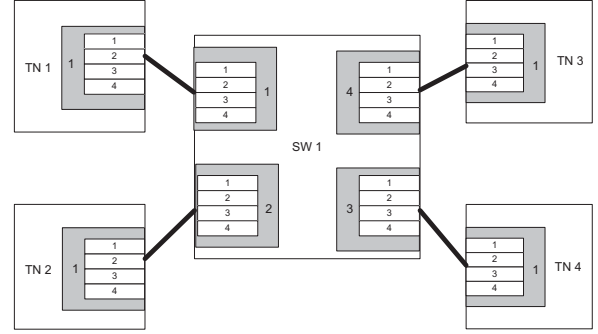


Fig. 3 Network model 1

at various details level – from general virtual components to cycle-accurate models of particular devices. Simulated device models are written in C++ [9].

In this case we used the router and node models which comprise only the Virtual Channel and the Network Layers (this gave an opportunity to reduce the simulation time and to obtain more detailed results). The link bandwidth in the model is set to 1 Gbit/s.

The results of the simulation can significantly depend on the router model implementation features such as local clock frequency and link capacity within the router.

##### C. Estimation of achievable characteristics of the Network model 1

Let us consider the case when each virtual channel has its own particular priority level, which corresponds to the virtual channel number: VC1 – the highest priority, VC4 – the lowest. The packet length does not exceed the frame length. Fig. 4-Fig. 11 shows the simulation results for the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> way of router implementation for each virtual channel, when size of data packet is 250 byte. Fig. 12 - Fig. 19 shows the simulation results for the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> way of router implementation for each virtual channel, when size of data packet is 750 byte.

2<sup>nd</sup> implementation of network layer differs by a large value of delay of high priority packet.

Fig. 12 - Fig. 19 shows the simulation results for the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> way of router implementation for each virtual channel, when size of data packet is 750 byte. Delay is bigger for the 2<sup>nd</sup> way of the router implementation than for 1<sup>st</sup>, 3<sup>rd</sup> way.

Fig. 20 - Fig. 23 shows the simulation results for the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> way of router implementation for each virtual channel, when size of data packet is 750 byte. There are interruptions the transmission of data with lower priority in 3<sup>rd</sup> way.

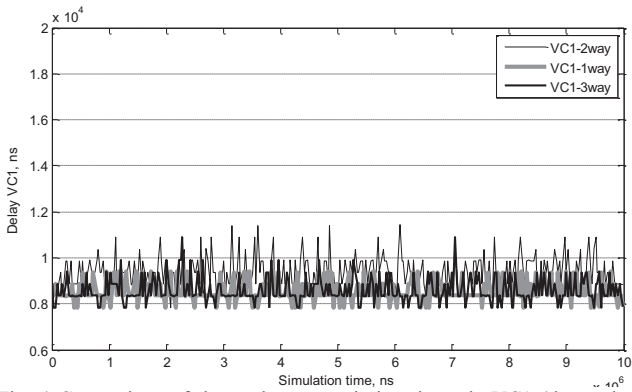


Fig. 4 Comparison of the packet transmission time via VC1 (the packet size = 250 bytes) in case of different implementations of network layer

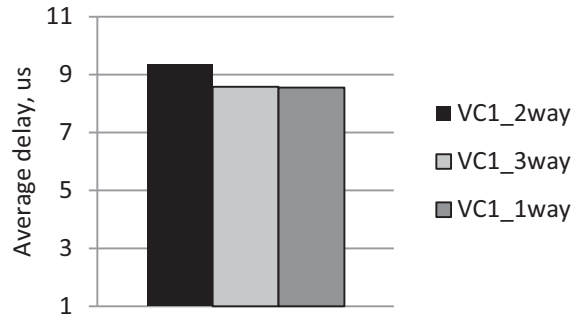


Fig. 8 Bar chart of the average packet transmission time via VC1 (the packet size = 250 bytes)

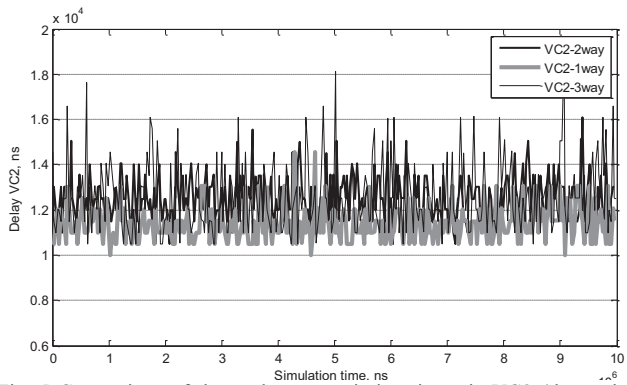


Fig. 5 Comparison of the packet transmission time via VC2 (the packet size = 250 bytes) in case of different implementations of network layer

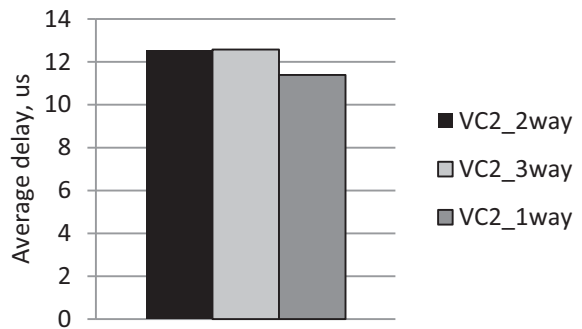


Fig. 9 Bar chart of the average packet transmission time via VC2 (the packet size = 250 bytes)

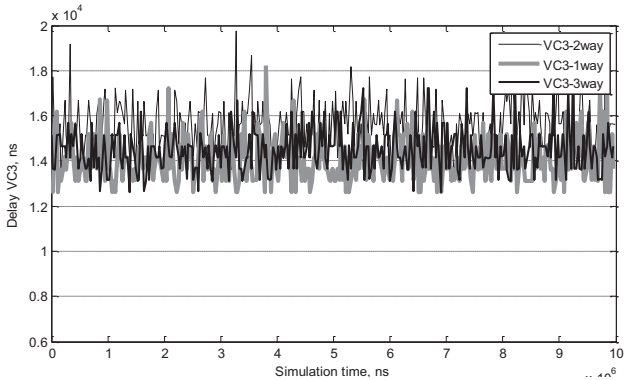


Fig. 6 Comparison of the packet transmission time via VC3 (the packet size = 250 bytes) in case of different implementations of network layer

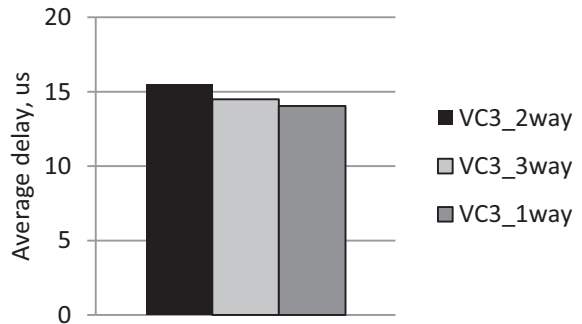


Fig. 10 Bar chart of the average packet transmission time via VC3 (the packet size = 250 bytes)

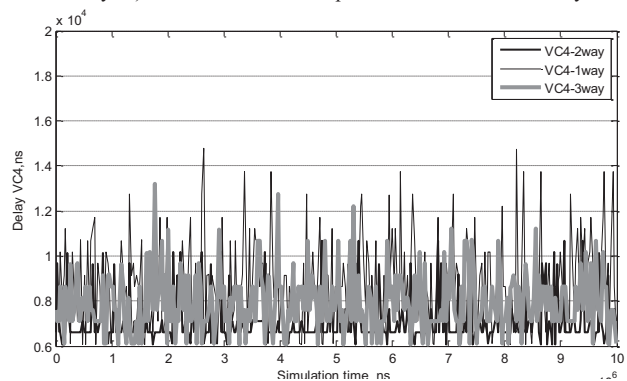


Fig. 7 Comparison of the packet transmission time via VC4 (the packet size = 250 bytes) in case of different implementations of network layer

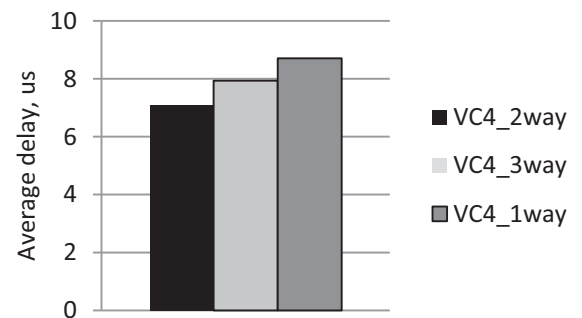


Fig. 11 Bar chart of the average packet transmission time via VC4 (the packet size = 250 bytes)

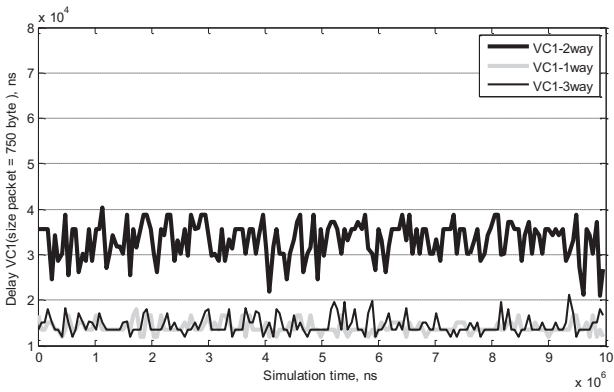


Fig. 12 Comparison of the packet transmission time via VC1 (the packet size = 750 bytes) in case of different implementations of network layer

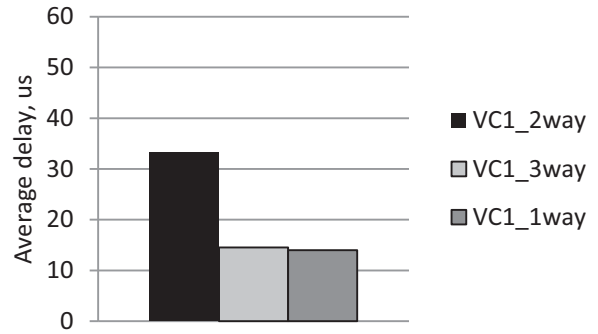


Fig. 16 Bar chart of the average packet transmission time via VC1 (the packet size = 750 bytes)

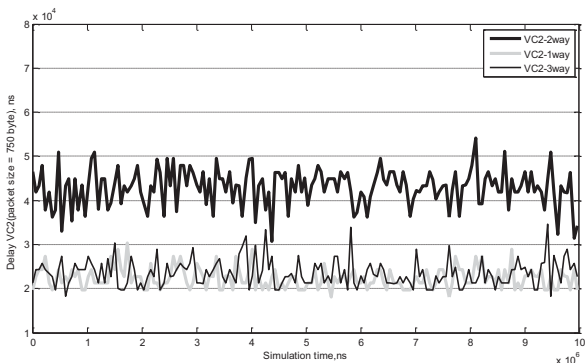


Fig. 13 Comparison of the packet transmission time via VC2 (the packet size = 750 bytes) in case of different implementations of network layer

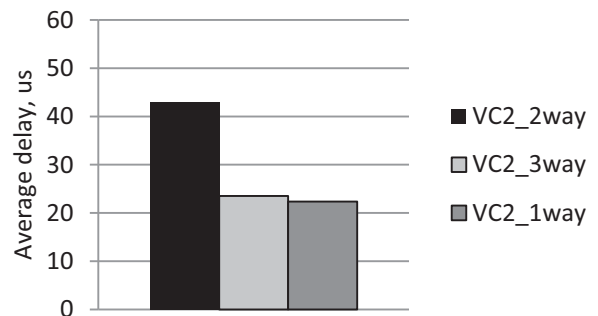


Fig. 17 Bar chart of the average packet transmission time via VC2 (the packet size = 750 bytes)

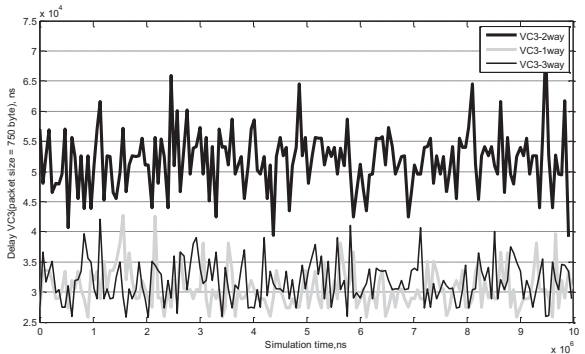


Fig. 14 Comparison of the packet transmission time via VC3 (the packet size = 750 bytes) in case of different implementations of network layer

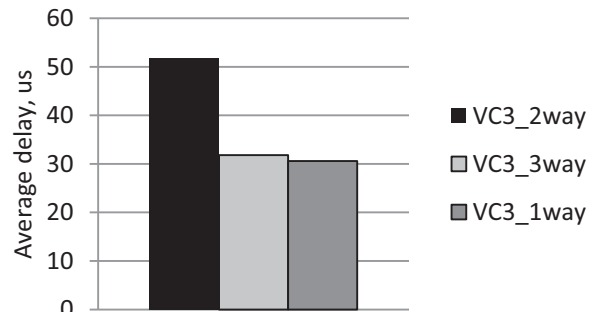


Fig. 18 Bar chart of the average packet transmission time via VC3 (the packet size = 750 bytes)

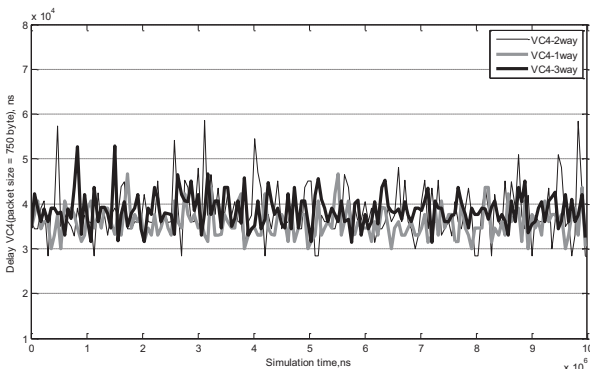


Fig. 15 Comparison of the packet transmission time via VC4 (the packet size = 750 bytes) in case of different implementations of network layer

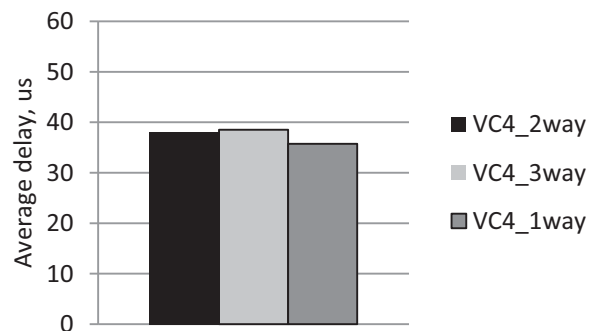


Fig. 19 Bar chart of the average packet transmission time via VC4 (the packet size = 750 bytes)

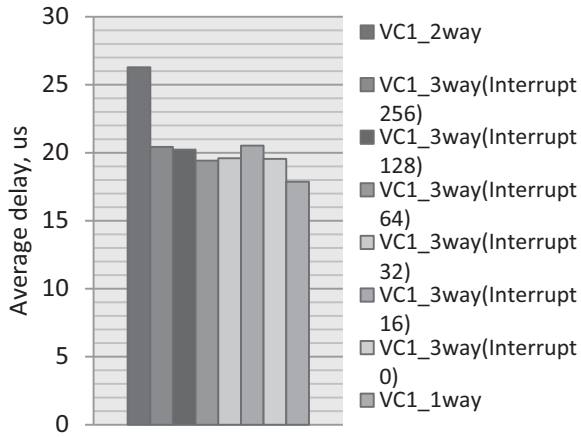


Fig. 20 Bar chart of the average packet transmission time via VC1 (the packet size = 750 bytes). Exponential distribution of packet generation time.

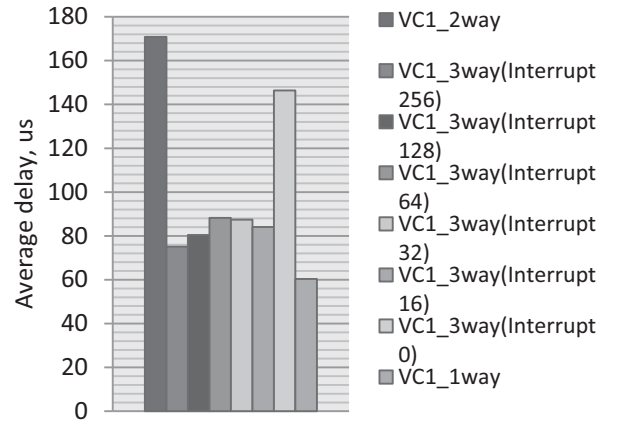


Fig. 23 Bar chart of the average packet transmission time via VC4 (the packet size = 750 bytes). Exponential distribution of packet generation time.

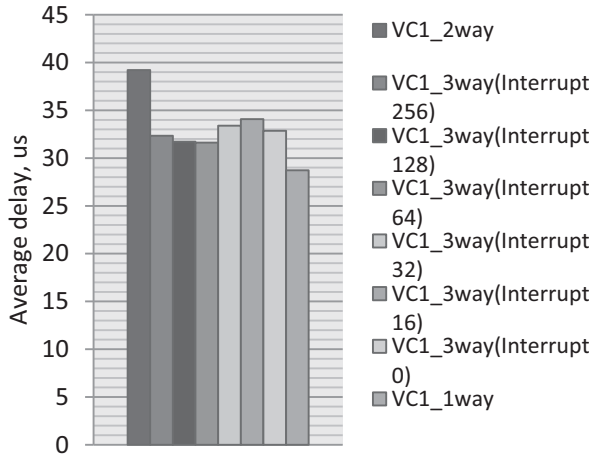


Fig. 21 Bar chart of the average packet transmission time via VC2 (the packet size = 750 bytes). Exponential distribution of packet generation time.

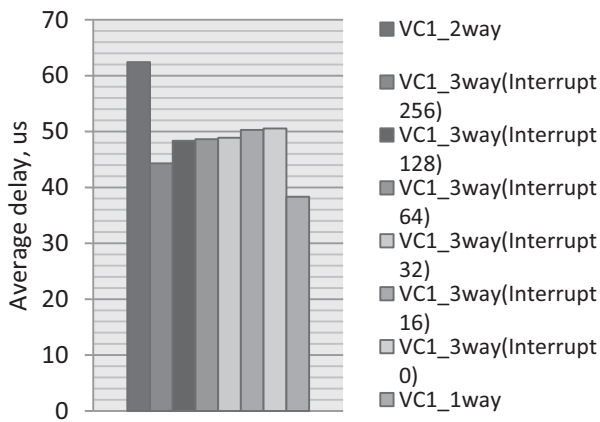


Fig. 22 Bar chart of the average packet transmission time via VC3 (the packet size = 750 bytes). Exponential distribution of packet generation time.

V. COMPARISON THEORETICAL AND SIMULATION RESULTS

Minimum and maximum delay for packet of the virtual channel with the highest priority was calculated using represented formulas. The parameters value:  $sizeP_{VC_h} = 250\text{byte}$ ;  $TwrByteRX_k = TwrByteTX_k = 8\text{ ns}$ ;  $sizeF = 256$ ;  $TcalcPrec_k = 16\text{ ns}$ ;  $DelaySwMatrix_k = 16\text{ ns}$ .

We calculate parameters of delay:  
 $minDelaySource_{VC_1} = 250 \cdot 8 + 16 = 2016\text{ ns}$ ;  
 $minDelayDestin_{VC_1} = 250 \cdot 8 = 2000\text{ ns}$ ;  
 $minDelaySw_k_{VC_1} = 250 \cdot 8 + 16 + 16 + 250 \cdot 8 = 4032\text{ ns}$ ;  
 $minDelay_{VC_1} = 8048\text{ ns}$ ;  
 $maxDelaySource_{VC_1} = 250 \cdot 8 + 255 \cdot 8 + 16 = 4056\text{ ns}$ ;  
 $maxDelayDestin_{VC_1} = 250 \cdot 8 = 2000\text{ ns}$ ;  
 $minDelaySw_k_{VC_1} = 250 \cdot 8 + 16 + 16 + 255 \cdot 8 + 3 \cdot (250 \cdot 8 + 16) = 10120\text{ ns}$ ;  
 $maxDelay_{VC_1} = 4056 + 2000 + 10120 = 16176\text{ ns}$ .

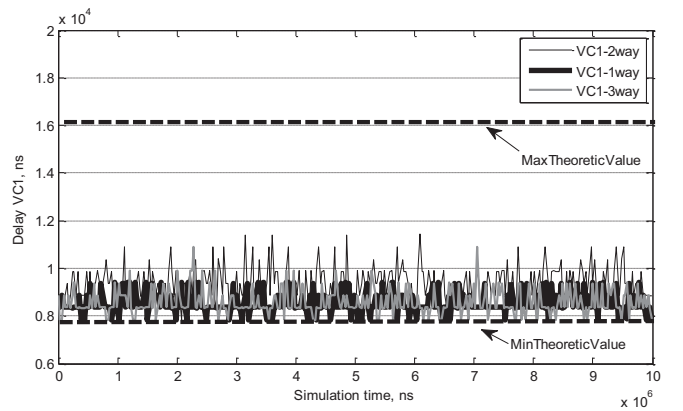


Fig. 24 Comparison of the packet transmission time via VC1 (the packet size = 250 bytes) in case of simulation and theoretical results

Theoretical minimum delay for packet of the virtual channel with the highest priority is equal simulation results. Theoretical maximum delay is more simulation delay. This is because the system is not functioning with maximum download.

VI. HARDWARE COSTS

We are using Cadence RTL Compiler and Encounter and UMC 120 nm technology library for evaluation of router’s switch matrix hardware cost. We performed a logical and a physical synthesis of the switch matrixes with different number of channels that correspond to different router implementations (different amount of ports and connection points).

Results of the logical synthesis are represented in Fig. 25. As shown in this figure, if quantity of connection points is bigger than 4, hardware cost grows essentially. The logical synthesis becomes impossible when quantity of ports is 16 and quantity of virtual channels is 16 or bigger (256 channels of the switch matrix). The physical synthesis is problematic if quantity of ports is bigger than 8 and of virtual channels is bigger than 8 (64 channels of switch matrix). This amount of switch matrix channels is boundary of hardware resources for the 1st way of a router structure. The 2nd way can be implemented with the greater amount of virtual channels if 2 – 4 connection point for every port is used. Thus the 1st way of a router structure hardware is essentially constrained.

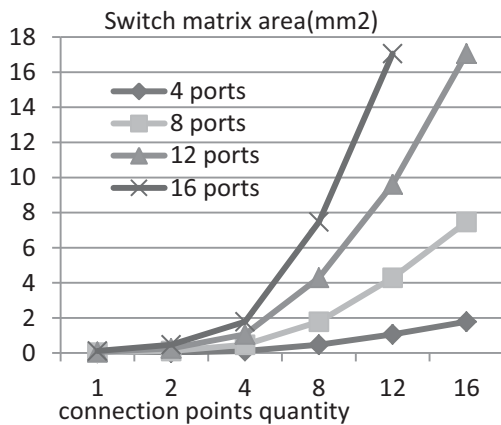


Fig. 25 The switch matrix hardware cost

VII. CONCLUSION

According to the investigations made the 1st way of the router organization results in the limitations in hardware implementation. The comparison of the achievable timing characteristics for different ways of router implementation showed that if a packet size is smaller than the frame size then the average packet transmission time for three ways is almost similar.

Delay of the low priority traffic grows faster for the 2nd way of the router implementation. Therefore, the 2nd way of

the router implementation can be used for the networks with the packet length shorter than frame size. In this case it will provide scheduled, bandwidth reserved and priority qualities of service. The packet lengths larger than the frame size while using the 2nd way of the router implementation result in degradation of the timing characteristics in comparison with the 1st and 3rd way. This degradation of characteristics grows proportionally to the packet’s length of the virtual channels of low priorities. Consequently, the 2nd way of the router implementation in networks where long packets are transmitted is possible only when there are no hard real time requirements. The 3rd way of the router implementation essentially decreases these disadvantages. The average packet transmission time and achievable link utilization in this case are almost similar to the 1st way of the router implementation.

Delay is 10% bigger for the 2nd way of the router implementation than for 1st way, when the packet length shorter than frame size and delay is 1% bigger for the 3rd way of the router implementation than for 1st way. Delay is 50% bigger for the 2nd way of the router implementation than for 1st way, when the packet length longer than frame size and delay is 17% bigger for the 3rd way of the router implementation than for 1st way. Therefore, the achievable characteristics for the scheduled service and delay value for this 2nd way of router implementation are lower.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation under grant agreement n° 13.G36.31.0003.

REFERENCES

- [1] S. Balandin, M. Gillet, “Embedded Network in Mobile Devices”, International Journal of Embedded and Real-Time Communication Systems (IJERTCS), Issue 1(1), pp 22-36. 2010.
- [2] D. Bueno, “Simulative analysis of the RapidIO embedded interconnect architecture for real-time, network-intensive applications”, in Local Computer Networks, 29th Annual IEEE International Conference, Nov. 2004, pp. 710 – 717.
- [3] S. Parkes, "SpaceWire for Adaptive Systems", ahs 2008 NASA/ESA Conference on Adaptive Hardware and Systems, pp.77-82.
- [4] J. Duato, S. Yalamanchili and L. Ni, *Interconnection Networks. An engineering approach*. San Francisco: Morgan Kaufmann Publishers, 2003.
- [5] J. Fleischmann, “Prototyping networked embedded systems”, *Computer.*, vol.32, Feb. 1999, pp. 116 – 119.
- [6] S. Parkes, A. Ferrer, A. Gonzalez, C. McClements, D3.2-SpaceWire-RT Updated Specification. Annex 1 SpaceFibre Standart, February 2013
- [7] W. James Dally, *Principles and Practices of Interconnection Networks*. San Francisco: Elsevier, 2004.
- [8] S. Parks, C. McClements, M. Dunstan, A. Ferrer, A. Gonzalez. SpaceFibre.SpW WG, October 2012
- [9] A. Eganayan, E. Suvorova, Y. Sheynin, A. Khakhulin, I. Orlovsky. DCNSimulator – Software Tool for SpaceWire Networks Simulation. International SpaceWire Conference 2013, June 2013