The Network Calculator for NoC Buffer Space Evaluation

Nadezhda Matveeva, Lev Kurbanov, Elena Suvorova, Yuriy Sheynin, Valentin Rozanov
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russian Federation
{nadezhda.matveeva, lev.kurbanov}@guap.ru, {suvorova, sheynin}@aanet.ru, valentin.rozanov@guap.ru

***Abstract*—In this paper we discuss the problem of choosing the buffer size in the Network-on-Chip routers. This problem is closely related to other problems that arise in NoC's design – choosing of interconnection structure between nodes and data paths in the system. It is a complex multicriteria problem. The design space exploration approach is widely used to solve such problems. In this approach each possible system configuration corresponds to a point in the Design Space. For each point, the user evaluates whether it satisfies its requirements and determine the future direction of motion in Design Space. The network calculators are used to calculate values of the NoC's parameters at each point. We consider the existing methods of buffer sizes calculation, their capabilities and limitations. We suggest the method of buffer space calculation for NoC with arbitrary topology and the algorithm of the corresponding network calculator.**

I. INTRODUCTION

Buffers are widely used in modern Network-on-Chip (NoC). They can apply in systems using packets routing/switching with buffering or with wormhole routing. For the latter case they allow to avoid data deadlocks and to decrease local overload inside some network area. It is achieved due to packets are accumulated in switch buffers inside such overloaded local network area. The distribution of local overload occurs much more slowly in the systems with buffering in comparison with the system that does not include buffers. Reduction of overloaded local network area can be provided by increasing buffers size. Designer should choose buffers size according to custom requirements of an average data transmission time across the system.

However hardware cost and energy consumption are essentially increased with buffer size growing. Hardware cost and energy consumption are high critical parameters of modern NoC. There are strong constraints for maximum buffer size in each switch. This limit depends on technology used for system implementation. Overall chip area and power constraints limits possible buffers area also.

Buffers size determination is complex and multicriteria task in System-on-Chip and Network-on-Chip design. This task should be solved in correspondence with topology selection and data distribution inside the NoC.

Design space exploration approach is widely used for solving such goals. A point in the design space corresponds to one possible variant of system implementation. This point characterized by configuration parameters and achievable performance characteristics of system. For our case, configuration parameters are NoC topology, buffer's size, data flow characteristics. Estimated performance characteristics are average data transmission time for each data flow, probabilities of buffers overload. When a system designer knows such combination of characteristics, he can decide, does the system correspond to custom requirements or not. If the performance characteristics are not satisfactory, then designer can define new configuration parameters and calculate other decision/point inside the design space.

Network calculators are popular tools for determination of NoC design space points. This group of methods is based on the use of analytical methods for calculation of NoC timing characteristics [1- 4]. Present network calculators can be divided into the following groups: deterministic; physical; based on queuing system; probabilistic.

The Deterministic approaches are based on graph theory. The model, developed by this method, can submit as cycle-static dataflow graph, which is used for buffer dimensioning for NoC applications. To show the effectiveness of this model, the analytical results were compared with those extracted from the simulation. However, deterministic approaches assume that the designer have a deep knowledge about the pattern of communication among cores and switches [3]. This leads to high computational complexity in implementation.

The Stochastic approaches are based on queuing theory. Usually for the calculation of queuing systems there uses the simple flows. There takes the simple flow, because it creates the most difficult mode of operation of the system as compared to other flows and is used to determine the limit values of quality of service characteristics. If the real flow is different from the simple, the system will work better than calculated. The advantage of this approach is a well-studied mathematical apparatus. The disadvantage is that the model is designed for the worst case, there is no possibility to take into account the actual characteristics of the data streams.

The Physics approaches are based on statistical physics and information theory to study the buffers behavior. Statistical physics can model the interactions among various components while taking into consideration the long-term memory effects. The main concept in this model is that packets in the network move from one node to another in a manner that is similar to particles moving in a Bose gas and migrating between various energy levels as a result of temperature variations. More precisely, the approach involves a virtual random growing network, which describes the NoC buffers evolution as a function of the packet injection rate. Performance analysis of the proposed model predicts that the buffer occupancy follows a power law distribution [3]. The upside is fairly accurate result analysis model. The disadvantage is time-consuming and computational complexity.

The System approaches are based on system theory. They are successfully applied to design electronic circuits. In particular, Network Calculus is inspired from this theory for modeling and evaluating the buffers size of each switch. The attractive feature of Network Calculus is its ability to capture all traffic patterns with the use of bounds. Based on shapes of the traffic flows, designers are able to capture some dynamic features of the network. Network Calculus can be used to analyze and evaluate performance metrics of onchip interconnects. It was demonstrated that Network Calculus theory can be used to find buffers size of each channel that best suit the traffic characteristics [3]. However, Network Calculus uses for worst-case analysis and can provide buffers depth and latency upper-bounds.

In this paper we present approach for system performance characteristics estimation, which depend on buffer size. Our approach is based on the queuing systems theory. This mathematical tool was chosen, because it allows to estimate the length of queues with minimal computational complexity.

Paper consists of 3 main sections. In the 1st section we consider features of existing analysis methods, limitations and dependences between buffer size and average data transmission time. In the 2nd section we present our approach. In the 3rd section the algorithm and its usage example are described.

II. THE EXISTING METHODS FOR THE BUFFER SPACE CALCULATION

In [2] the authors propose the methodic of buffer space distribution for mesh network with wormhole routing algorithm. The main idea of the algorithm – is finding the busiest channels and expansion of buffer space in routers, where the probability of buffer overflow is the highest. The size of others, underutilized buffers is decreased correspondingly. Summary size of all buffers in all routers is constant.

The work presented in [3] is extended from the modeling approach for macro computer network analysis and evaluation by considering also the processing rate at each router. This concept of processing rate was also used in Network calculus method in which the maximum data flow, which can be sent by each switch, is constrained by the arrival flow and its average processing rate. Authors mainly show how the compartmental Fluid-flow theory can be used for studying and analyzing the non-linear behavior of NoC applications. In particular, the Fluid-flow model is used as a design space exploration methodology for customizing or tuning buffers size of on-chip interconnect architectures given a particular application traffic pattern.

In [4] authors proposed to introduce a decoupling buffer (D-buffer) in a target IP. This buffer receives data from the NoC with jitter, while the target IP consumes data from this buffer at the application rate, without jitter. Authors propose a general method to define D-buffer size and threshold, considering influence of packaging, arbitration, routing and concurrency between flows. Application traces highlight the threshold parameter importance.

In [5] authors address an optimization problem of minimizing the total number of buffers subject to the performance constraints of the applications running on the SoC. Also, authors formulate another optimization problem to minimize the variances of buffer size in the respective output buffers of switches. As both of the mentioned objective functions are worthwhile for the design process, authors formulate them as a multiobjective problem under the QoS constraints.

In [6] authors present an algorithm to find the minimal decoupling buffer sizes for a NoC using TDMA and credit-based end-to-end flow-control, subject to the performance constraints of the applications that run on the SoC.

The existing methods have limitations - they do not allow applying them for NoCs with arbitrary topology. Many of them focus also on a specific flow control mechanism, service discipline.

III. THE PROPOSED SCHEME OF BUFFER SPACE CALCULATION

We use the queuing systems theory to calculate the temporal characteristics of the system depending on the characteristics of the data streams and buffer sizes.

The input data for the calculation are the network topology, the distribution of data streams, buffer sizes in routers. In accordance with these input data is performed the calculation of average transmission time of data packets between sources and receivers. There is defined a point in the Design Space, that corresponding to this set of input data. This point can or cannot comply with the requirements of the user.

The user can change the system configuration if it does not satisfy his requirements. He may repeat calculations until an appropriate configuration will be found. Reconfiguration of the system in design can include increasing of buffer space. If the maximum permissible size of buffers is reached, then the network topology or the data transmission paths should be changed.

The current version of network calculus is designed for the NoC with arbitrary topology; buffers are located in input ports of routers, the non preemptive and non priority service of packets is used, the adaptive routing is not used.

The received from the user network topology (an example is represented in Fig. 1) is converted to the network queuing system (the example represented in Fig. 2). The nodes of the queuing system correspond to input port controllers of the routers with buffers. Every node is considered as separate queuing system. Since the physical size of the buffer is limited we will consider the queuing systems with limited queue length, type M/M/1/N. We use a simple data stream, as it creates the most difficult mode of the system load in comparison with other flows. It is used to determine the worst values of characteristics. If the actual flow is different from the simple one, the system will work better than calculated.

In considered NoCs the packets are never discarded; therefore density of the input data packets flow is equal to density of the output data flow for all queuing systems.

$$\lambda_{in} = \lambda_{out}$$

The calculation of the network parameters must be done from the end to the beginning, i.e. from the receiver to the source. This scheme is typical for most network calculus. It allows to calculate the size of the queue at the node, which is located close to the source, in accordance with the delay that is added by located closer to the receiver nodes. Let's evaluate a calculation scheme for the example network shown in Fig. 1, Fig. 2. In this example two data flows came up in the "Switch Node 1 Port 3 out".

An example of the NoC topology fragment is in Fig. 1. Switches and terminal nodes are represented by rectangles on this figure. The data flows between source and receiver nodes are showed by arrows. Terminal nodes call "Source node" and "Receiver node". Each node has some ports. Each port in each node has output and input parts.

To calculate the parameters of represented NoCs topology, we need to convert this scheme to queuing network. An example of the corresponding network queuing system is shown on Fig. 2. The buffer is in the input part of the port, therefore we represent this point as queuing system, type M/M/1/N. "Service" and "Buffer" are elements of the queuing system, where "Buffer" is queue, "Service" is servicer. "Conflict point" is the point through which the multiple flows pass. Exactly the same notations are valid for Fig. 4.

We call this node, in which multiple data flows compete, the conflict point. This point can be in one of three states, described by next probabilities: P_0 - the probability of the idle state (no packets pass); P_1- the probability of the data flow 1 transmission (the probability of the data packet from first data flow transmission); P_2 - the probability of the data flow 2 transmission (the probability of the data packet from second data flow transmission). The sum of these probabilities is equal to one. The probability of the concrete data stream transmission is equal to load ρ of the queuing system by this data stream.

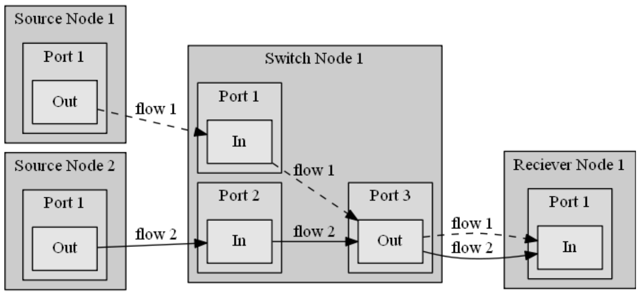

Fig. 1. An example of the NoC topology fragment

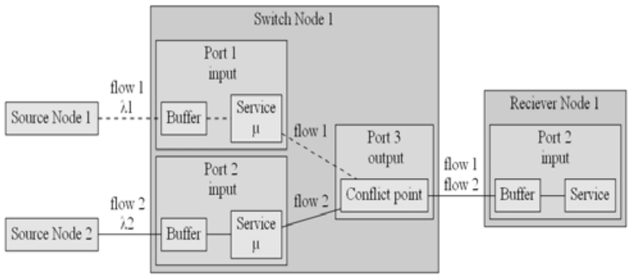

Fig. 2. The corresponding network queuing system

Let's consider the node "Switch Node 1 Port 1 input". The probabilities of all possible states of the system are:

- the idle state P_0^1;
- the data transmission state P_{trans}^1;
- the accumulation of data in the buffer state P_{accum}^1;
- the buffer overflow state P_{over}^1;

Based on the foregoing for "Switch Node 1 Port 1 input" the probability of the data transmission state is

$$P_{trans}^1 = P_1 \quad (1)$$

The probability of the data flow 1 transmission is

$$P_1 = P_{LoadCN} \cdot (1 - P^2_{trans}), \tag{2}$$

where

$$P_{LoadCN} = \frac{\sum_{i=1}^{n} \lambda_i}{\mu_{LoadCN}}. \tag{3}$$

The node “Switch Node 1 Port 1 input” is in the accumulation of data in the buffer state when the node “Switch Node 1 Port 3 out” is in the data flow 2 transmission state.

$$P^1_{over} + P^1_{acm} = P_2 \tag{4}$$

The probability of the buffer overflow state can be evaluated by next formula:

$$P^1_{over} = \rho^N \frac{1-\rho}{1-\rho^{N+1}}, \tag{5}$$

where

N – the maximal quantity of the packets in system

ρ – the load of system

$$\rho = \frac{\lambda}{\mu}. \tag{6}$$

Let’s evaluate the packet transmission time via current node

$$W = T + T_{delay}, \tag{7}$$

where

T – the packet transmission time if the packet is not retained in the buffer

T_{delay}- the retaining time if the next point (closest to destination) is busy by another data flow.

$$T = \frac{PacketSize}{\mu} \tag{8}$$

$$T_{delay} = \frac{\sum_{i=1}^{N-1} P_i \cdot i \cdot t}{N-1} \tag{9}$$

This calculation scheme can be used for every node in the network queuing system.

IV. THE NETWORK CALCULATOR DESCRIPTION

In this section we present detail information about proposed algorithm for NoC buffer space evaluation. The main steps of algorithm are network parameters loading, calculation probabilistic characteristics for each data flow in each network node, calculation timing characteristics for each data flow transmission. The scheme of algorithm is shown on Fig. 3.

Input data describing network should contain list of network nodes and their functional characteristics, list of links between nodes, list of data flows. Every data flow is characterized by the data transmission path between source and destination node, packet arrival rate and packet size.

The transmission path of data flow comprises data about each network node, which transfers it. Functional characteristics of network nodes and links are throughput, ports number.

When we have full knowledge about network and data flows, then we can analyze where and which data flow are competed for transmission medium. Moreover we have opportunity to estimate workload in each network node. It is useful for getting information about reserves and bottlenecks of system design. All steps of algorithm are proposed below.

We shall describe the proposed network calculator on the case study. At the beginning network configuration parameters are loaded from description files. Description files contain information about network topology, data flows parameters, buffers size, services intensity. Data flows parameters are data intensity, data transmission paths. After that, we select one of the receivers. For example, let us choose “Receiver node 1”.

TABLE I. PARAMETER NOTATION

Paramet er	Description
λ	The packet arrival rate
μ	The packet service rate
N	The maximal quantity of the packets in system
ρ	Load of system
P^x_0	Probability of the idle state for x node
P^x_{trans}	Probability of the data transmission state for x node
P^x_{acm}	Probability of the accumulation of data in the buffer state for x node
P^x_{over}	Probability of the buffer overflow state for x node
P_{LoadCN}	Probability of the load conflict point
T	The packet transmission time if the packet not retained in the buffer
T_{delay}	The retaining time if next point (closest to destination) is busy by other data flow.
W	The packet transmission time via current node

```
Algorithm 1 Calculate transmission time of the package
Step 0
    Load the network parameters and form the lists of sources and receivers.
Step 1
    If the list of receivers is empty, then go to step 7, else
    - Calculate the characteristics of the input stream of packets from all sources for the next node from the list of receivers.
    - Determine the workload and packet processing time.
    - After calculating, remove the node from the list of receivers.
Step 2
    Perform step towards the source. Find all children of the calculated in step 1 node.
    If the child node is the conflict point and has multiple parents (if the conflict point is an input port), then go to step 3.
    If child node is the conflict point and has children (if the conflict point is an output port), then go to step 4.
    If child node does not conflict, then go to step 5.
    If there are not any children, then proceed to Step 1.
Step 3
    Calculate all parents of the conflict point. Then calculate probabilistic and temporal characteristics of the conflict point.
Step 4
    Calculate the load of the conflict point. Calculate the probability of dominance for each flow.
Step 5
    Calculate the temporal characteristics of the node, taking into account the probability at the node-parent.
Step 6
    Repeat from step 2 until calculate all the nodes in the network.
Step 7
    Determine the transmission time of each flow through the system along the entire route of transmission. In this step, the point is formed in the design space of possible solutions.
Step 8
    If the results do not satisfy the user requirements or the user needs to calculate network with the new sizes of the buffers, then repeat the algorithm with the new system characteristics.
```

Load of system is calculated by equation 6. “Receiver node 1” is the final node, which delay transmission time is equal to 0. Therefore, processing time in this node does not depend on buffer size and we should use equation 8. “Receiver node 1” is deleted from receiver list after its parameters calculation.

Next step is selection a new node in the receiver list according to data transmission path. We define all children of “Receiver node 1”. Let's assume that output ports don’t have buffers so calculation of their timing parameters don’t make sense. Node “Switch node 1 Port 2 output” isn’t conflict point, thus it is enough to calculate its probabilistic characteristics. The node “Switch node 1 Port 1 input” is a child of the node “Switch node 1 Port output”.

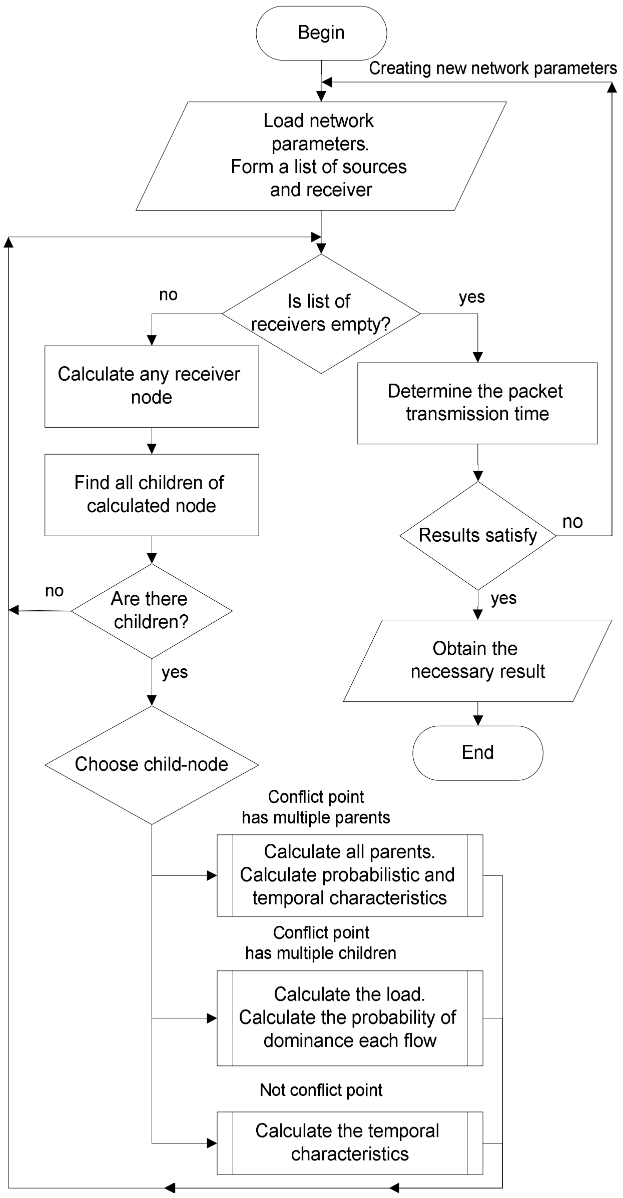
Begin
Creating new network parameters
Load network parameters. Form a list of sources and receiver
Is list of receivers empty?
no
yes
Calculate any receiver node
Find all children of calculated node
Are there children?
no
yes
Choose child-node
Conflict point has multiple parents
Calculate all parents. Calculate probabilistic and temporal characteristics
Conflict point has multiple children
Calculate the load. Calculate the probability of dominance each flow
Not conflict point
Calculate the temporal characteristics
Determine the packet transmission time
Results satisfy
no
yes
Obtain the necessary result
End

Fig. 3. Scheme of proposed algorithm

This node is the conflict point, because more than one data flow goes via it. At the same time, it represents the input port of the router and therefore has a buffer space. Then we check presence of several parents or several children for this node. The node “Switch node 1 Port 1” has three parents: “Receiver node 1”, “Receiver node 2”, “Receiver node 3”. We have already calculated the parameters for “Receiver node 1”. Now we need to compute parameters for “Receiver node 2” and “Receiver node 3”.

In case where processing rate μ is identical for all devices, presence of the buffer will not matter, and the processing time is calculated by the formula (8). If μ for different nodes are different, there is a possibility of data accumulation in the buffer of “Switch node 1 Port 1 input”. In this case we need to consider the probability of buffer overflow in parent nodes to calculate the packet transmission time.

Then we take a step in the direction of the source. We are looking for children for “Switch node 1 Port 1 input”. The child node is “Switch node 2 Port 3 output”. This node is also a conflict point, but it corresponds to output port of the router. Therefore we calculate only the probabilistic characteristics for them. This node has only one parent and two children. Then we evaluate the transmission probabilities for all data flows with using the formulas (2) and (3).

Then we take next step toward the source – chose an arbitrary child node. For example, we select “Switch node 2 Port 1 input”. This node is not a conflict point as only one data flow goes via it. We calculate probability and timing characteristics taking into account the probabilities obtained in the parent node.

On the next step we reach the source node. Now we can calculate the average packet transmission time for the first data flow. We remove current source node from the list of source nodes, because it is the source for only this data flow. Since the list of sources is not empty, we should return to the point, where the parent had several children. This point is the node “Switch node 2 Port 3 output”. After it we choose the next child node and perform similar calculations. We shall get a point in the Design Space when the source list will be empty. If the results are not satisfactory, or it is possible to improve them, the user can load new network parameters and calculate again.

V. Conclusion

The task of buffer space development is important for modern systems-on-chip (SoC) and NoCs. It determines the NoC area and power dissipation and also affects the system performance and quality of service provided by such systems.

In this paper we have present the developed Network Calculator for the NoC buffer space evaluation. It allows to calculate values of the NoC’s parameters at a point in the Design Space.

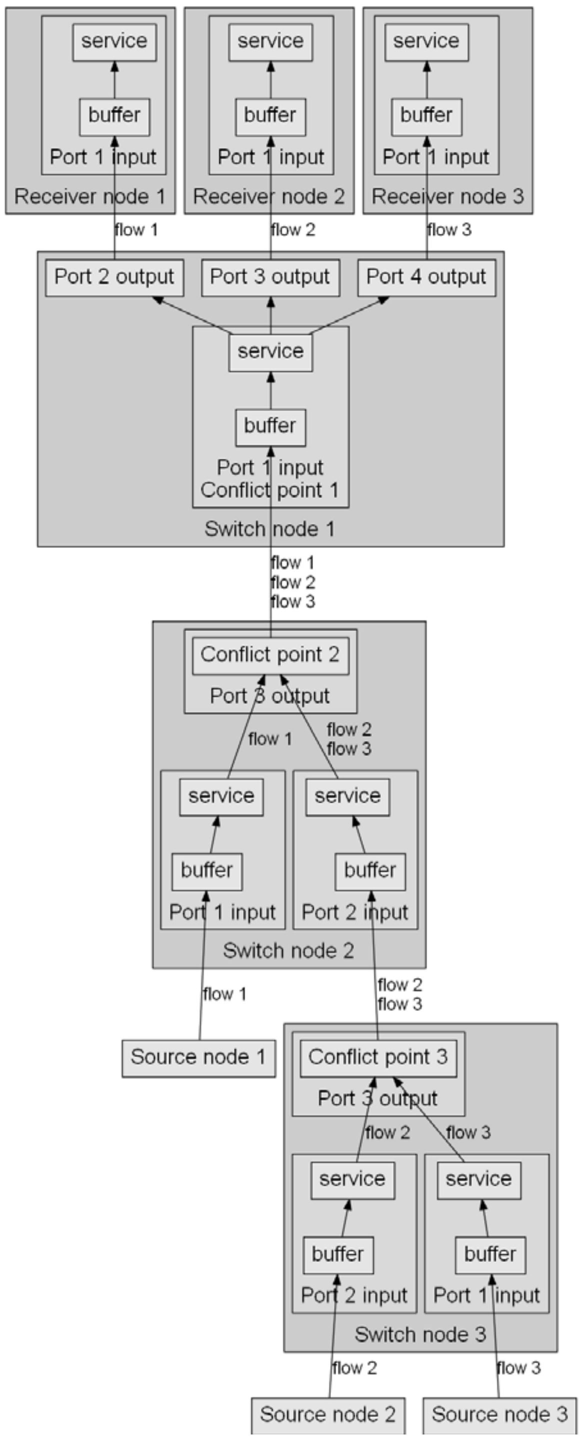

Fig. 4. Example of the network structure

Proposed Network Calculator for the NoC buffer space evaluation is useful at the design stage. Its results help to check features of NoC before implementation. If developed system doesn’t satisfy custom requirements, we can simply change input parameters and get new point in the Design Space. The Network Calculator can be used for networks with arbitrary topology, without any constraints. The buffer sizes in different input ports of different routers can be various. By the proposed calculator we can design networks with optimal buffer size, which are able to provide a reliable and required data transfer.

In this paper we have present the developed Network Calculator for the NoC buffer space evaluation. It allows to calculate values of the NoC's parameters at a point in the Design Space. Proposed Network Calculator for the NoC buffer space evaluation is useful at the design stage. Its results help to check features of NoC before implementation. If developed system doesn't satisfy custom requirements, we can simply change input parameters and get new point in the Design Space. The Network Calculator can be used for networks with arbitrary topology, without any constraints. The buffer sizes in different input ports of different routers can be various. By the proposed calculator we can design networks with optimal buffer size, which are able to provide a reliable and required data transfer.

Our directions for further work are supporting:

- multiple data priority levels;
- virtual channels with different buffer size for each virtual channel;
- adaptive routing;
- possibility of buffers in output ports.

ACKNOWLEDGMENT

The research leading to these results has received financial support from the Ministry of Education and Science of the Russian Federation according to the base part of the state funding assignment in 2015, project no. 1810.

REFERENCES

[1] M. Bakhouya, S. Suboh, J. Gaber, T. El-Ghazawi, S. Niar, "Performance Evaluation and Design Tradeoffs of On-Chip Interconnect Architectures, Simulation Modeling Practices and Theory", *Simulation Modelling Practice and Theory*, vol. 19, no. 6, June 2011, pp. 1496-1505.

[2] L. Wang, Y. Cao, X. Li, and X. Zhu, "Application specific buffer allocation for wormhole routing networks-on-chip," *NoCarc08, MICRO-41*, 2008.

[3] M. Bakhouya, A. Chariete, J. Gaber, M. Wack "A Buffer-space Allocation Approach for Application-specific Network-on-Chip", *Computer Systems and Applications (AICCSA), 2011 9th IEEE/ACS International Conference on*, 2011, pp. 263-267.

[4] Leonel Tedesco, Ney Calazans, Fernando Moraes, "Buffer Sizing for QoS Flows in Wormhole Packet Switching NoCs", *Proceedings of the 20th Annual Symposium on Integrated Circuits and Systems Design*, 2007

[5] Fahimeh Jafari, Zhonghai Lu, Axel Jantsch, Mohammad Hossein Yaghmaee, "Buffer Optimization in Network-on-Chip Through Flow Regulation", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Dec. 2010,pp. 1973 – 1986.

[6] M. Coenen, Eindhoven Philips Res., K. Goossens, G. De Micheli, S. Murali, M.Coenen, "A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control", *in Proceedings of the 4th International Conference Hardware/Software Codesign and System Synthesis*, Oct. 2006, pp. 130 – 135.