NoC Performance Parameters Estimation at Design Stage

15th Conference of Open Innovations Association FRUCT

Nadezhda Matveeva, Elena Suvorova
State University of Aerospace Instrumentation
Saint-Petersburg, Russia

Introduction

- The task of designing effective systems is difficult
- Weight, power consumption of Network-on-Chip (NoC) should be as low as possible
- Performance, processing speed should be as high as possible
- Different methods of performance evaluation can be used when the structure of the system is being designed

Current calculators of performance evaluation

- Deterministic network calculators
  - Based on the graph theory
- "Physical" network calculators
  - Based on usage of calculator models built for various physical processes
- Network calculus
  - Based on flow models for characteristics calculation
- Probabilistic calculators
  - Based on the theory of queuing systems

Proposed network calculator

- Based on the queuing system
- Can be used for different types of communication systems (bus, switch, NoC)
- Can be used to evaluate the performance of communication systems based on different topologies (regular and irregular)
- Supports networks with wormhole routing and with buffering

Calculation principles

- Open-loop stochastic networks are used to perform evaluations
- Communication system considered as a set of servers and queues of requests for servers
- Requests are transactions between applications running on master and slave devices
- The fact is taken into account that different transactions between different master and slave devices may have different timing values

Main steps

Step 0

- At the initial stage data path between communicating applications *i* and *j* is defined. Number of data flows is defined for each switch.

Step 1

- Each switch is assigned unique number. Data transmission time for each flow is computed for each of transit switches. Total load of the output port is determined.

Step 2

- Average and maximum time for each flow is calculated along all data paths. Switch port load is calculated.

Formulas

Taken into account:

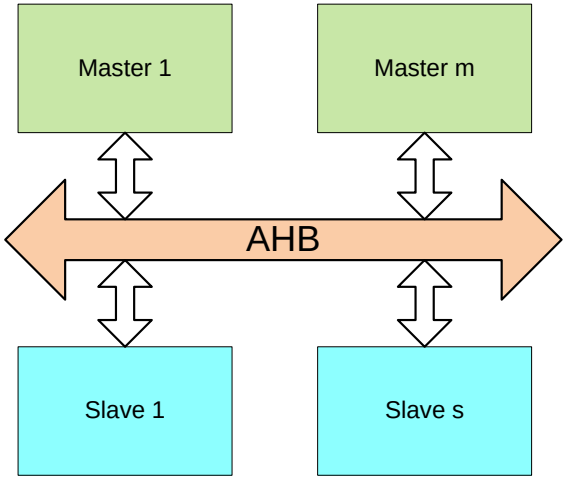
- Communication system type (bus, switch, NoC)
- Routing scheme (wormhole routing and with buffering)
- Type of service disciplines (with priority and without)
- Slave's ability to perform reading and writing transactions in parallel

Some definitions

Ts	Transaction transmission time
Tt	Time interval between application requests
W	Average waiting time
$Tsum$	Total transmission delay of data flows through concrete port
Th_k	Header processing time in switch k
$Trech_j^i$	Packet header transfer time from the input port to the output port of the switch
$Ttrans_j^i$	Time of packet data transmission through switch output port
$Trec_j^i$	Data transmission time of packet between application i and j through switch input port
R	Load

AHB Bus

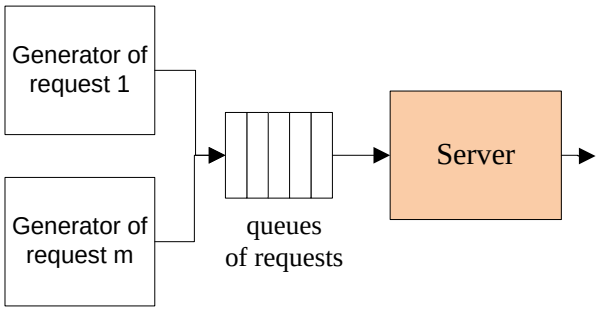
- AHB bus has 1 server for read and write transactions
- Bus load is calculated taking into account all transactions



$$R_{ij} = \begin{cases} \frac{Ts_{ij}}{Tt_{ij}}, Tt_{ij} > 0, Ts_{ij} > 0 \\ 0, Tt_{ij} = 0 U Ts_{ij} = 0 \end{cases}$$

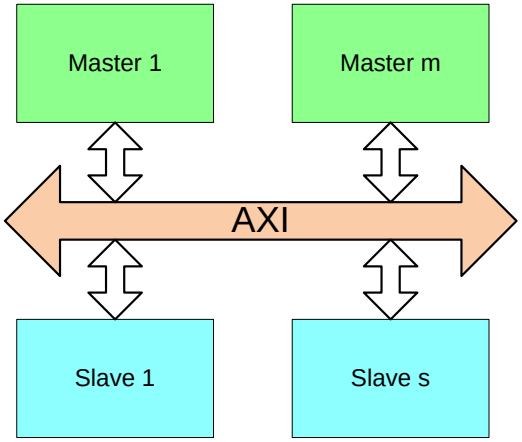
$$Rbus = \sum_{i=0}^{|\{Mm\}|-1} \sum_{j=0}^{|\{Ms\}|-1} R_{ij}$$

$$Tavij = W_{ij} + Ts_{ij}$$



AXI Bus

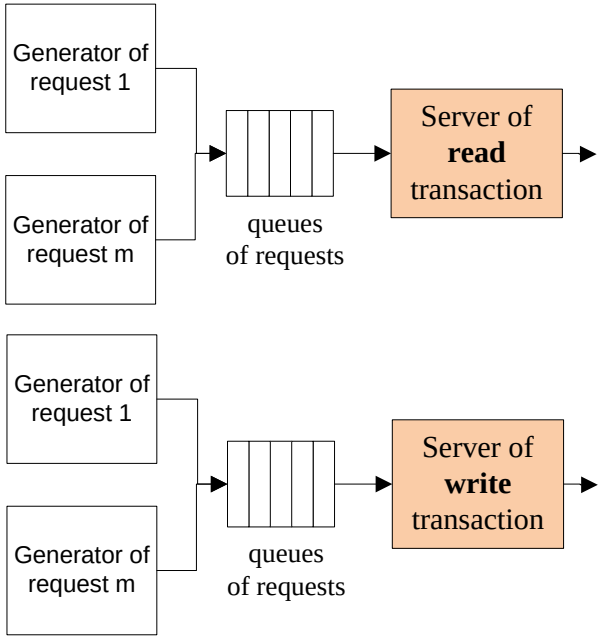
- AXI bus has 2 separate servers (server of read and write)
- Read and write bus load are calculated separately



$$Rr_{ij} = \begin{cases} \dfrac{Tsr_{ij}}{Ttr_{ij}}, Ttr_{ij} > 0, Tsr_{ij} > 0 \\ 0, Ttr_{ij} = 0 UTsr_{ij} = 0 \end{cases} \qquad Rw_{ij} = \begin{cases} \dfrac{Tsw_{ij}}{Ttw_{ij}}, Ttw_{ij} > 0, Tsw_{ij} > 0 \\ 0, Ttw_{ij} = 0 UTsw_{ij} = 0 \end{cases}$$

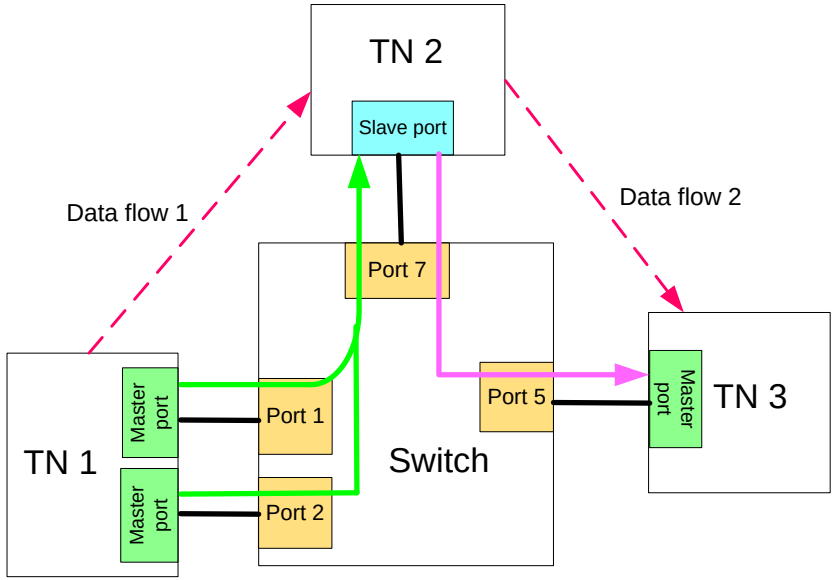
$$\mathrm{Rread} = \sum_{i=0}^{|\{Mm\}|-1} \sum_{j=0}^{|\{Ms\}|-1} \mathrm{Rr_{ij}} \qquad \mathrm{Rwrite} = \sum_{i=0}^{|\{Mm\}|-1} \sum_{j=0}^{|\{Ms\}|-1} \mathrm{Rw_{ij}}$$

$$\mathrm{Tavr_{ij}} = \mathrm{Wr_{ij}} + \mathrm{Tsr_{ij}} \qquad \mathrm{Tavw_{ij}} = \mathrm{Ww_{ij}} + \mathrm{Tsw_{ij}}$$



Switch

- Uses a hierarchical queuing system
- Servers of slave devices are on the first level of hierarchy
- Servers of master devices are on the second level of hierarchy
- Switch port is server



first level

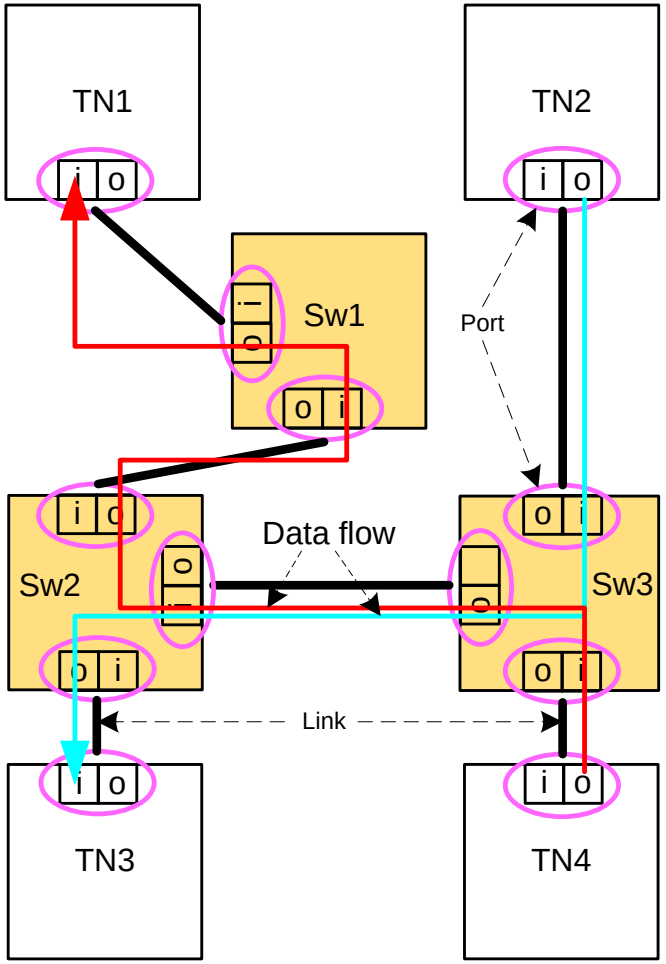
$$Rs_{ij} = \begin{cases} \frac{Ts_{ij}}{Tt_{ij}}, Tt_{ij} > 0, Ts_{ij} > 0 \\ 0, Tt_{ij} = 0 UTs_{ij} = 0 \end{cases} \qquad Rs_{l} = \sum_{i=0}^{|\{Mm\}|-1} \sum_{j=0}^{|\{Ms_l\}|-1} Rs_{ij} \qquad Us_{lij} = W_{lij} + Ts_{ij}$$

second level

$$Rm_{ij} = \begin{cases} \frac{Us_{lij}}{Tt_{ij}}, Tt_{ij} > 0, Ts_{ij} > 0 \\ 0, Tt_{ij} = 0 UTs_{ij} = 0 \end{cases} \qquad Rm_{l} = \sum_{i=0}^{|\{Mm_l\}|-1} \sum_{j=0}^{|\{Ms_l\}|-1} Rm_{ij}$$

NoCs

- Server is input/output part of a port
- For all switches data flows are defined according to routing information
- Load of output part of the port is calculated as total utilization of all data flows passing through this output port
- Average data transmission time for each data flow is estimated in every port of route and along the entire route



Some formulas for NoC

Average and Maximum transmission packet delay between application *i* and *j* through transit *k* switch (**wormhole routing**)

$$Tavij_k = Trech_j^i + Th_k + W + Ttrans_j^i$$

$$Tmax_j^i = Trech_j^i + Th_k + Tsum + Ttrans_j^i$$

Average and Maximum transmission packet delay between application *i* and *j* through transit *k* switch (**with buffering**)

$$Tavij_k = Trec_j^i + Th_k + W + Ttrans_j^i$$

$$Tmax_j^i = Trec_j^i + Th_k + Tsum + Ttrans_j^i$$

Build data path

Step 0
Represent the system in the graph form. Device of the system is vertex. Link between adjacent nodes is edge. All vertexes have the "unmarked" status.

Step 1
Lists "Front1" and "Front2" are empty. Specify the source and receiver vertexes.

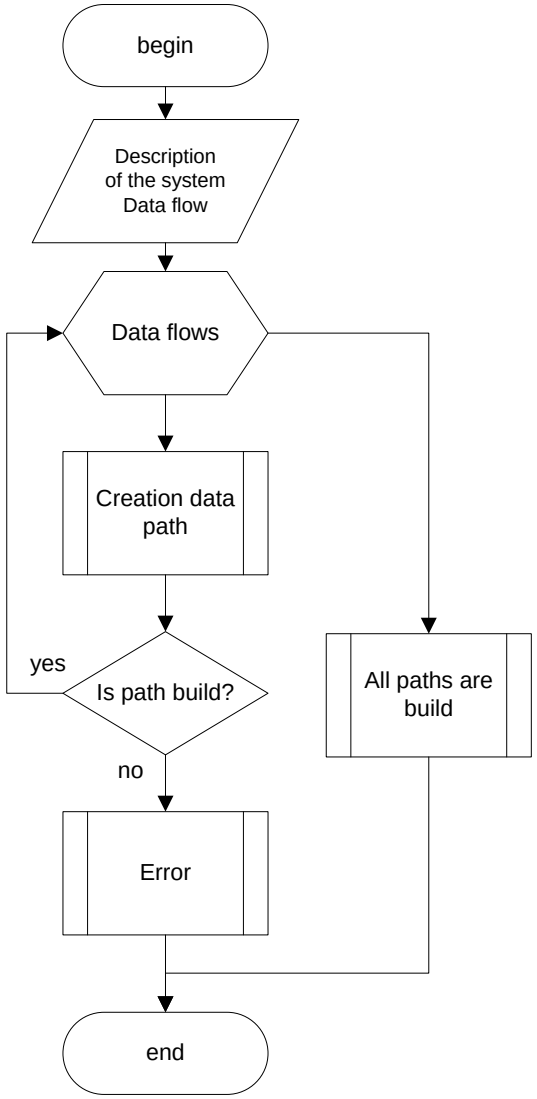
Step 2
Front1 = source vertex

Step 3
While the receiver vertex is not reached or the list Front2 is empty perform *step 4 - step 5*

Step 4
Front2 = Front2 + adjacent vertex from the set of vertexes with "unmarked" and "marked" status.

Step 5
For all vertexes from Front1 status is "viewed". Clear Front1. Copy Front2 to Front1. For all vertexes from Front1 status is "marked".

Step 6
If receiver vertex is achievable, then data path is build.
If receiver vertex is not achievable, then data path cannot be built.

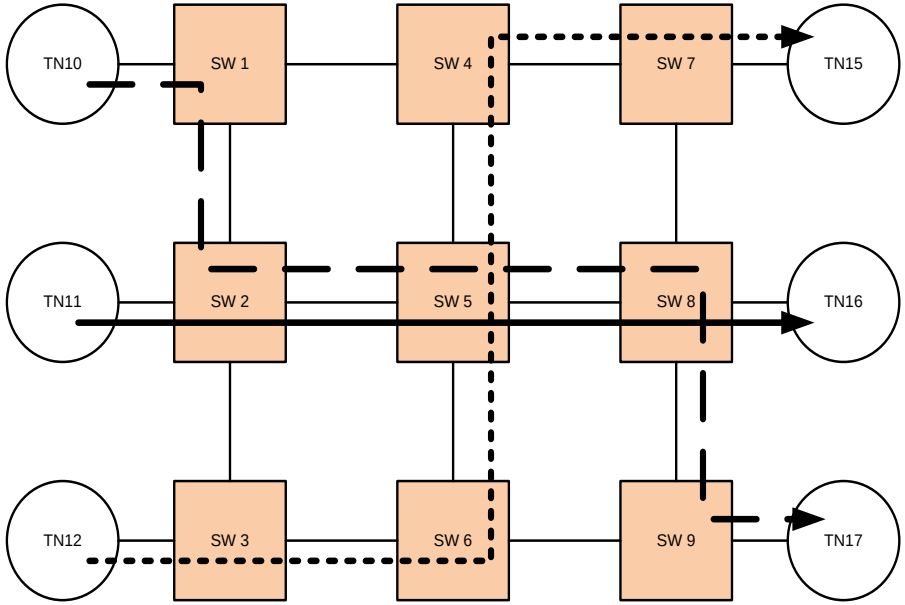


Modeling System

- We use the Digital Communication Network Simulator (DCNSimulator)
- It is based on Qt and SystemC
- It consists of the simulation engine and libraries of network components
- Simulated device models are written in C++
- Application software algorithms can run at end nodes thus generating realistic traffic for the simulated network

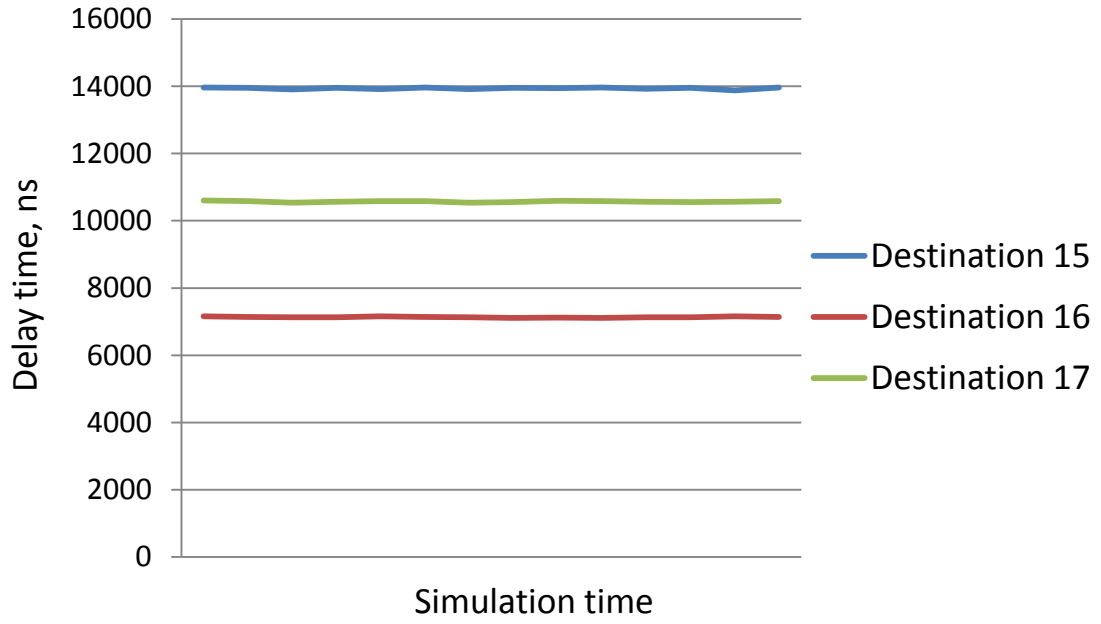
Example1

- Type of communication system is NoC
- This system consists of 6 terminal nodes and 9 switches
- Network topology is mesh
- Header length = 2 byte
- Packet length for all data flows = 126, 254, 510 byte
- Clock period = 10 ns
- Header processing time in switch = 35 clock cycles



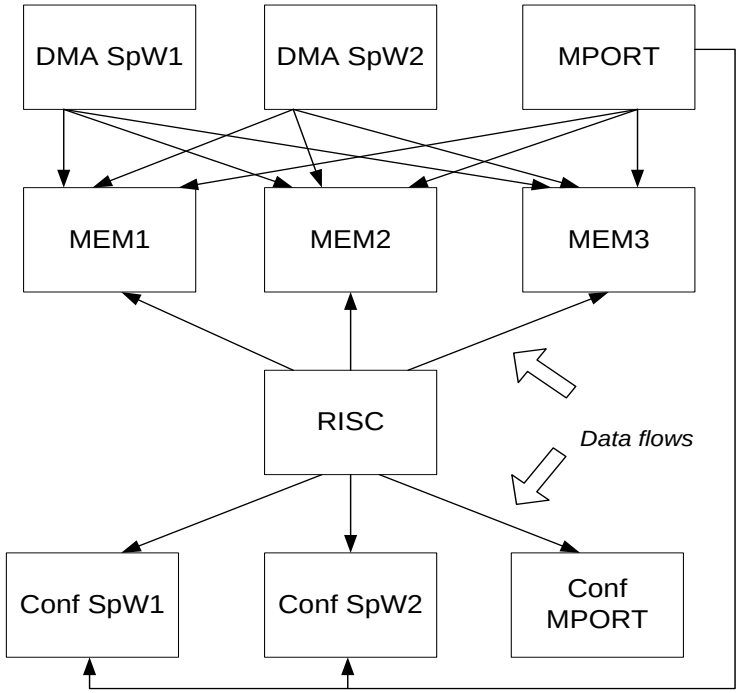
Simulation and Theoretical Results

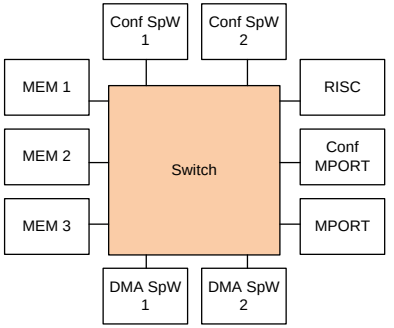
Data flow (source-destination)	Theoretical maximum transmission delay, ns	Simulation maximum transmission delay, ns
TN10–TN17	11000	10610
TN12–TN15	16300	13960
TN11– TN16	9200	7160



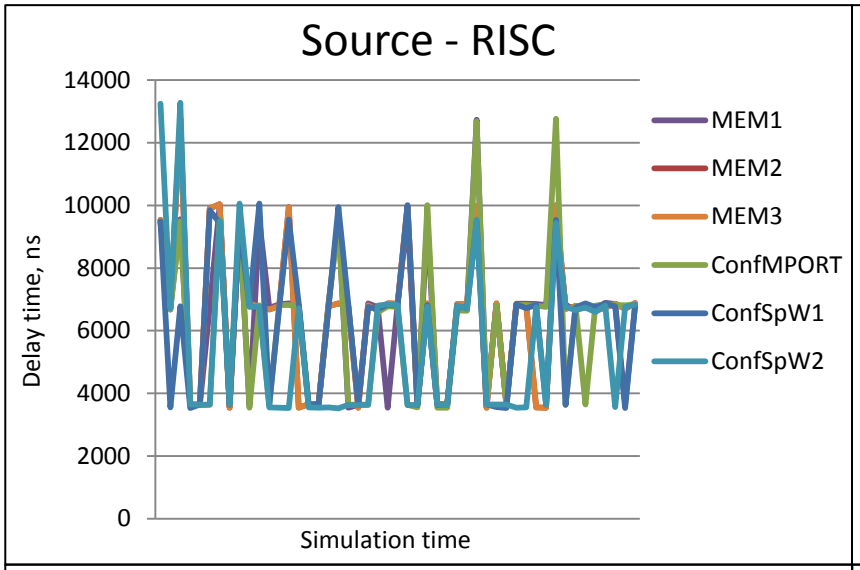
Example2

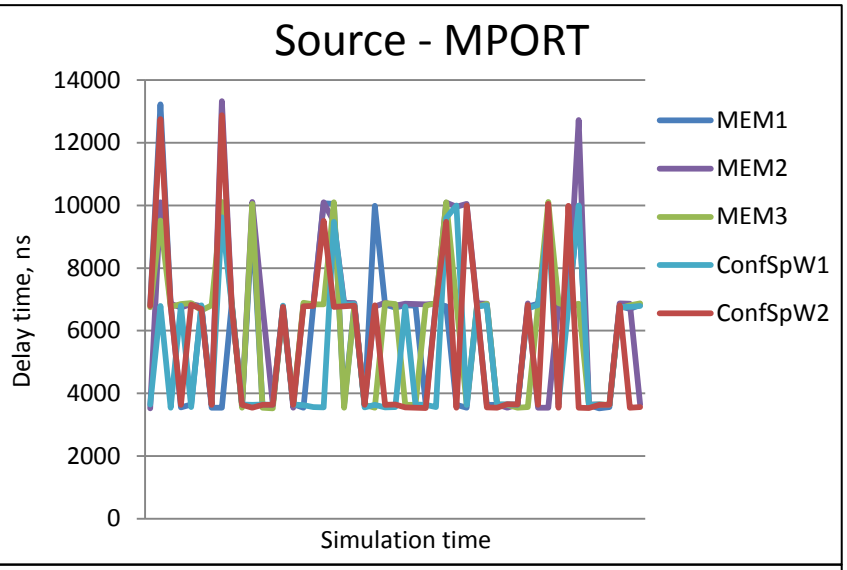
- Type of communication system is switch
- This system consists of RISC, memory blocks (MEM1, MEM2, MEM3), two DMASpW and ConfSpW, MPORT and ConfMPORT
- Header length = 2 byte
- Packet length for all data flows = 126 byte
- Clock period = 10 ns
- Header processing time in switch = 35 clock cycles

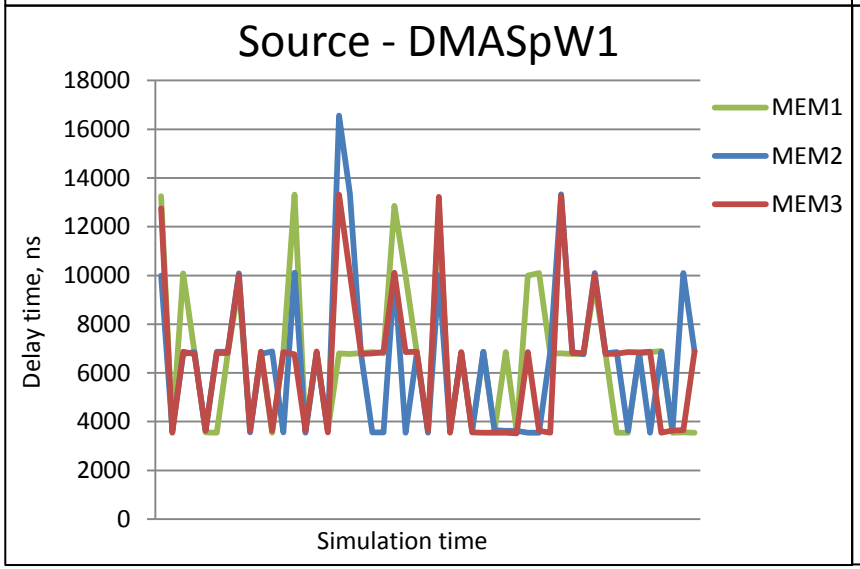


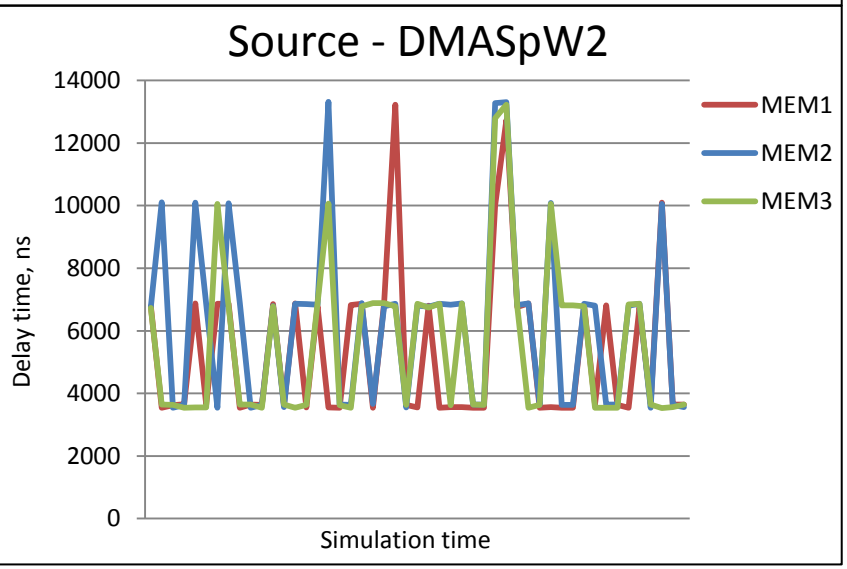


Simulation results









Results

Data flow (source-destination)	Theoretical maximum transmission delay, ns	Simulation maximum transmission delay, ns
RISC–MEM3	20700	13270
MPORT–MEM3		16420
DMASpW2– MEM3		13290
DMASpW2–MEM1		15920
RISC–MEM1		13280
DMASpW1–MEM1		**16500**
MPORT–MEM1		13330
DMASpW1–MEM3		16480
RISC–MEM2	24200	13280
RISC–ConfSpW2		13270
MPORT–MEM2		13350
DMASpW1–MEM2		**16560**
DMASpW2–MEM2		13330
RISC–ConfSpW1	13700	**13230**
MPORT–ConfSpW1		13220
RISC–ConfMPORT	13200	12760
MPORT–ConfSpW2		**12900**

Conclusion

- We proposed an approach to evaluate the performance and operating characteristics of different communication systems
- Using the obtained values of system characteristics designer can evaluate the system performance at the stage of architectural design
- It allows identifying the bottlenecks in the system and verifies that the system corresponds to requirements according to which it was developed
- Proposed method can also be used for networks with wormhole routing and with buffering

Contact information

- **Matveeva Nadezhda**

  e-mail: n.matveeva88@gmail.com
- **Suvorova Elena**

  e-mail: suvorova@aanet.ru


ACADEMY
INDUSTRY
FRUCT


SUAI