Architectural Design of Grand Clos Collective Network for Supercomputers

PLAMENKA BOROVSKA, DESISLAVA IVANOVA
Computer Systems Department,
Technical University of Sofia,
Sofia
BULGARIA
pborovska@tu-sofia.bg, d_ivanova@tu-sofia.bg
http://cs.tu-sofia.bg/

Abstract: As modern research and engineering computing methodologies, the demands for high-speed computational resources are growing at a rapid rate. The communication performance of collective network is a crucial factor influencing the communication performance of supercomputers which form the prevailing parallel architecture of modern high-performance computer systems. In this paper we have proposed a modular high-speed switch architectural design and a new collective network design of “Grand Clos” topology to meet the demands of efficient and high-speed communications on supercomputers. The communication performance parameters such as network latency and throughput are evaluated on the basis of parallel simulation models using the framework OMNET++ (C++, MPI) and run on IBM HS22 Blade Center. Analysis and comparison of simulation results has been performed and presented considering the impact of the traffic pattern and the packet size on the communication efficiency of the collective network.

Key-Words: Collective Networks, Supercomputers, Grad Clos Architecture, Network Topology Design, OMNET++, Communication Performance Evaluation

1. Introduction

Switch and collective network architecture designs are significantly influenced by next generation high-performance computer systems and supercomputer technology. The path toward realizing next generation exascale and z-scale computer systems is increasingly dependent on building supercomputers with thousands of processors. The supercomputer architecture interconnects of collective topology is a crucial factor in determining the computer performance, [1-4].

Collective networks vary with respect to throughput, latency, scalability, and cost. Network performance determines supercomputer performance for many applications. Therefore, the initial choice of a collective network design will affect the usability and performance of supercomputers. Interconnection design of collective networks is composed of a set of shared switch nodes and channels, and the topology of the network refers to the arrangement of these nodes and channels.

Selecting the collective network topology design is the first step in designing a network because the routing algorithm and flow-control method depend on the topology. Selecting a good topology is the most important job of fitting the requirements of the collective network design to the available supercomputer technology [1], [3].

The goal of this paper is to suggest generalized and modular high-speed switch architectural design and a relevant collective network design of Grand Clos topology utilizing cut-through routing and to evaluate its communication efficiency for building up supercomputers. Communication performance of a high-speed switch and Grand Clos topology designs are performed by means of network simulations using OMNET++ and run on IBM Blade Center, located at High-Performance and GRID Computing Laboratory, Computer Systems Department, Technical University of Sofia.

2. OMNeT++ Framework

OMNeT ++ is essentially a set of software tools and libraries that supports the development of simulation models. Most often OMNeT++ is used to develop models of computer networks and protocols, but the product can be used for the preparation of various models. OMNeT++ represents simulation environment, including specific libraries (simulation framework and library). It is built up of individual components called modules. Its main purpose is to be used for building network simulations of various kinds, including wired and wireless communication networks, embedded networks (NoCs) and others. The functionality, specific to certain field as a simulation of Internet protocols, support for sensor networks, optical networks and others may be further implemented as a separate, independent project and built later in other projects. OMNeT++ includes Eclipse-based graphical development environment (IDE) and some additional tools to facilitate the work of the developers.

There are also extensions for real-time simulation, networking emulation, opportunity for using alternative programming languages such as Java and C# (not often used in making models), the possibility of integrating databases and many additional features, [5].

3. Switch Architecture

We suggest a switch architecture that has a highly regular structure. Every switch has its own address – a function from its coordinates. Main building blocks in our switch model are registers, buffers, multiplexers and output ports. The object input register is the one that incorporates the routing function. It is implemented with factory method pattern, so construction of exact routing method is done at initialization depending on configuration variable. This approach provides the ease of adding new routing algorithms. The same technique is used for traffic patterns and some other places as well for making the model extensible, [7], [8].

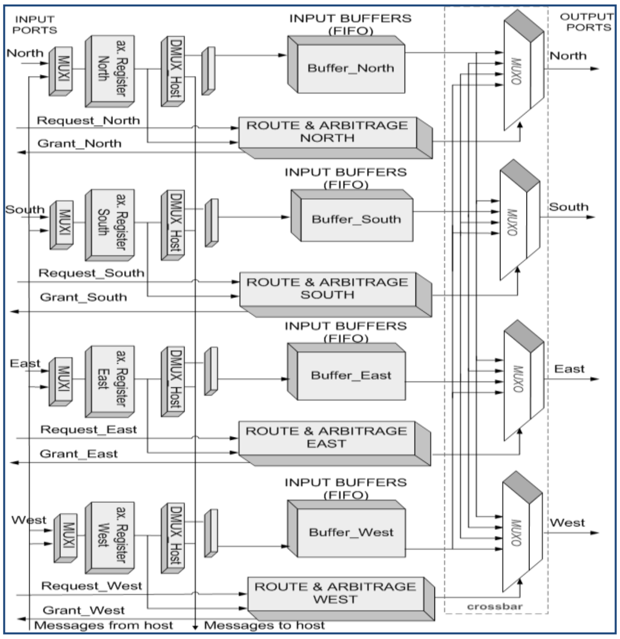

Fig. 1 Generalized Switch Design

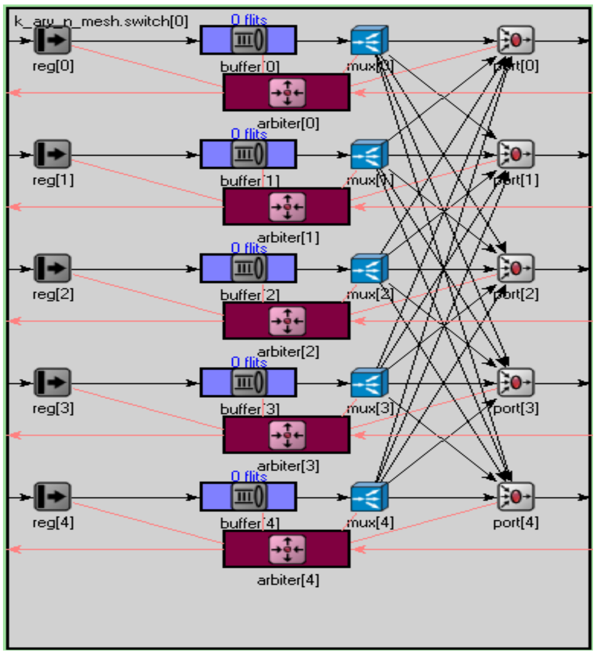

Fig. 2 Simulation View of the Switch Design

FIFO buffer is implemented as an object which size is a configurable parameter. Along with its data input and output, it also has a signaling output, which produces corresponding messages when the FIFO is either full or empty. Output ports are actually demultiplexers in data path and together with multiplexers implements a non-blocking crossbar in this implementation of the switch. Of course, these main modules are abstract enough and might be used for other switch designs. Links between them are provided by OMNET++ build-in objects DatarateChannel and their physical parameters are configurable. Control path is provided with separate links and allows different switch-to-switch (link level) flow controls – credit-based, on/off and ack/nack. This mechanism is provided by In Control and Out Control modules. Out Control is also responsible for switch and VC allocation.

4. Collective Network Architecture

We choose an architectural design of collective network for supercomputers based on its cost and performance. The cost is determined by the number and complexity of the chips required to implement the collective network, and the length of the interconnections on boards. Performance has two components: throughput and latency. Both of these measures are determined by factors other than topology, for example, flow control, routing strategy, and traffic pattern. To evaluate just the topology, we develop measures, such as bisection bandwidth and channel delay that reflects the impact of the topology on performance. Also, in most cases, the proper choice is combination of two or more topologies, combined in such way, that the network design combines their advantages. Therefore we suggest an innovative collective network topology design of Grand Clos, combining Fat Tree and Clos topologies, [6], [7], [9].

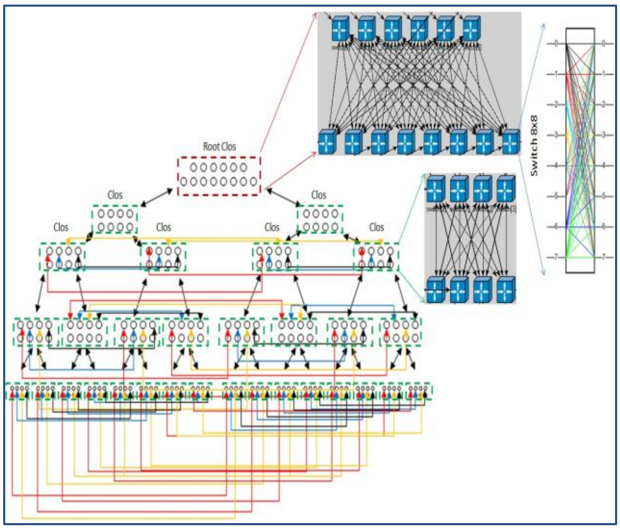

Fig. 3 Architectural Design of Grand Clos Collective Network

The proposed new GRAND CLOS architectural design is a hybrid design of Fat Tree and Clos, named Grand Clos. As a hybrid model, it has significant advantages, compared to the classic Clos and Fat Tree topologies. Actually, the new topology design consists of multiple interconnects of the Clos topology, connected in a Fat Tree network. The most important part of the topology is the horizontal links between the nodes of the fat tree on the same level. They minimize the needed hop count in order to communicate from one level to another. Also, these horizontal links reduce the load of the root node. Normally, it splits the Fat Tree network in two parts. Therefore, the entire traffic from the first part to the second one has to go through the root node. In our design the horizontal links help with the communication between the parts of the network, which leads to increased throughput and minimized latency. In addition, the link between the closses is reliable, because every node has connection to the others, based on a Crossbar. So, our idea is to use the advantages of both topologies in the best way in a hybrid model, with additional improvement – horizontal links. This will decrease the network latency, which is seen from the simulation results.

4.1 Topology Design

A single Clos interconnect consists of 8 nodes (2 rows, 4 nodes each, (Fig. 4). Each node from the top row is connected to all 4 nodes from the bottom row, as well as each node from the bottom row is connected to all nodes from the top row (these links are not shown for better readability). The root node (the Clos at the top of the tree) has 6 nodes on the first row and 8 nodes on the bottom one. The motive is the need of only downward links. It has 16 links to the second level (8 for the left branch and 8 for the right one). Each from the other Clos interconnects has 8 connections to the upward level, 16 to the downward one and 8 horizontal connections. So, the total count of nodes is 254 (30 small closses, 8 nodes each and 1 bigger clos (root clos) with 14 nodes).

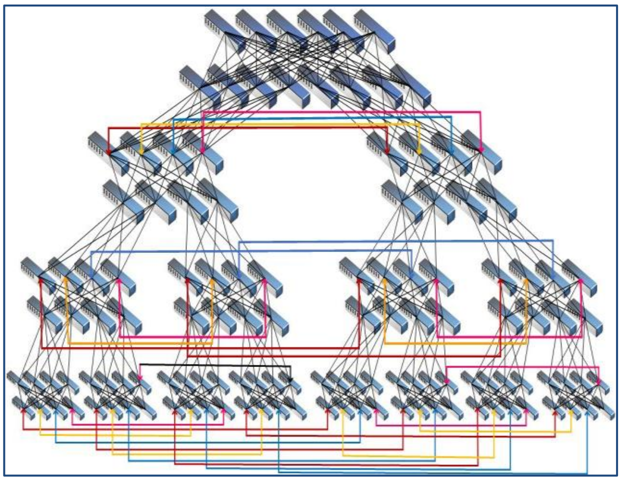

Fig. 4 Design of Grand Clos topology

Another important part of the proposed topology design is the horizontal links between closses on the same level. The team considers for each clos, nodes 2 and 4 (blue and black arrows) are connected to a different clos, as nodes 1 and 3 (red and orange arrows). This helps to reduce the hop count, while transmitting messages.

4.2 Modules

Root Clos: It has 6 switches on the first row and 8 on the second and there is no need for horizontal connections and therefore it consists of 14 switches in order to ensure the communication between the 2 parts of the network. It has 16 links – 8 to the left side and 8 to the right one.

Clos: There are 30 8-switch closses in total. Each such clos has 32 available links. 8 of them are used to connect to the clos on the higher level. 16

switches are connected with the 2 closses on the lower level. The remaining 8 connections are used for side connections to 2 other closses on the same level.

Switch8x8: Every switch has 8 input and 8 output ports. There are also input registers, buffers with capacity of 1 flit, which are connected to the output ports.

4.3 Topology Design Implementation – Author's Vision

After analyzing the design of the IBM's Blue Gene/P supercomputer, we find the proposed Grand Clos topology can be easily implemented with a similar hardware design.

In BlueGene/P, thirty-two compute cards and, optionally, up to two I/O cards are packaged onto the next-level board, called the node card. Sixteen node cards are plugged from both sides into a vertical mid-plane card, completing an assembly of 512 compute nodes in an 8 x 8 x 8 configuration. The inbound and outbound network connections for this 512-way cube are routed to four link cards that carry a total of 24 Blue Gene/P link (BPL) chips. The assembly of 16 node cards, 4 link cards, and an additional service card is called a mid-plane or a 512-way, [2].

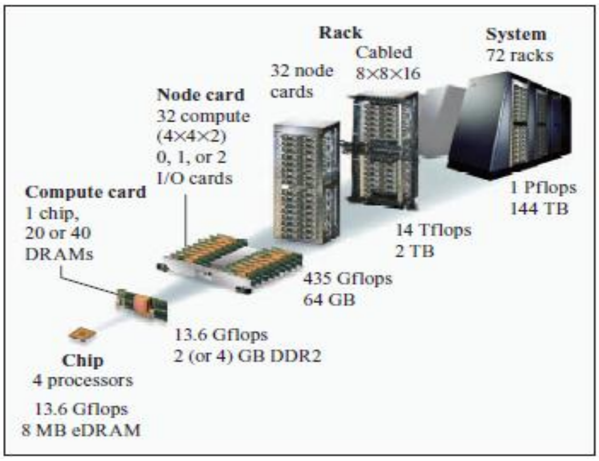

Fig. 5 Blue Gene/P design (Source: Blue Gene Red Book, p. 19)

The implementation of the topology can be achieved the following way:
All chips on a node card can be organized in a single clos. Then, 8 node cards, representing 8 closses on the same level in the network can be organized in one rack. The racks will be connected in a Fat Tree, following the topology design.

5. Parallel Model and Implementation in OMNET++

5.1 Experimental Platform

In order to provide a scaling of models, there is need for a lot of memory and CPU power. If the computer, on which the simulation is running, has insufficient performance, the simulation will take a lot of time. Therefore, the simulations are performed on the IBM Blade Center, located at the High-Performance and GRID Computing Laboratory, Computer System Department, Technical University of Sofia. The hardware platform of IBM Blade Center is based on Blade Server HS22, 2xXeon Quad Core E5504 80w 2.00GHz/800MHz/4MB L2, 3x2GB and three Blade servers, HS21, Xeon Quad Core E5405 80w 2.00GHz/1333MHz/12MB L2, 2x1GB Chk, O / Bay SAS to disk subsystem IBM System Storage DS3400 Single Controller and hard disk drive subsystem for IBM 750GB Dual Port HS SATA HDD chassis specialist for Blade Center, IBM eServer BladeCenter (tm) H Chassis and recorder 2x2900W PSU UltraSlim, network switch Blade Center Chassis , BNT Layer 2 / 3 Copper Gb Ethernet Switch Module, Optical Switch chassis specialist for Blade Center, Brocade (R) 10-port 4 Gb SAN Switch Module with Optical Switch Module for IBM Short Wave SFP Module, together with the necessary wiring, special cabinet Blade Center, NetBAY S2 42U Standard Rack Cabinet and Power Ultra Density Enterprise C19/C13 PDU Module (WW).

5.2 Full NED Usage Diagram

The following diagram shows usage relationships between simple and compound modules, module interfaces, networks, channels and channel interfaces.

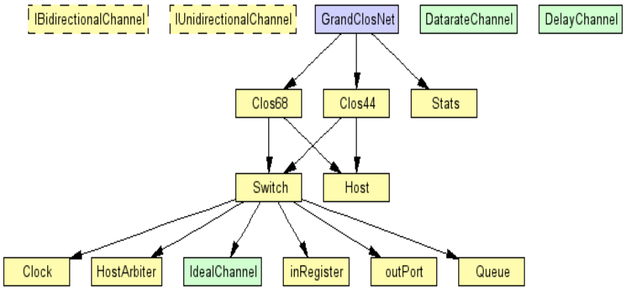

Fig. 6 OMNeT++ module hierarchy

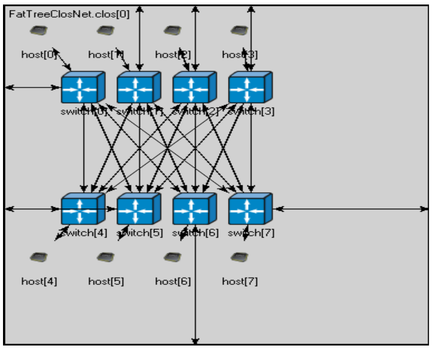

Fig. 7 OMNeT++ Clos Module

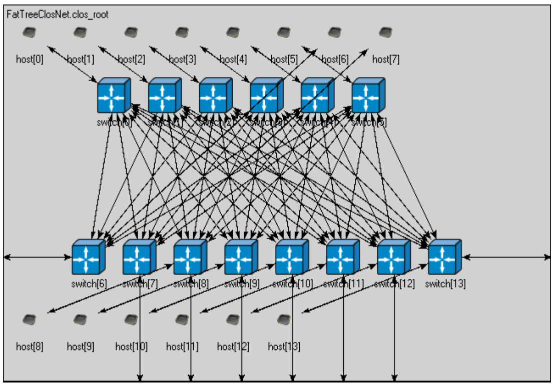

Fig. 8 OMNeT++ Root Clos Module

For each clos, the number of hosts is equal to the number of switches, so that one host is connected to every switch. Fig. 5.C shows the switch components. The clock module handles the Cmessages. Cmessages represent all events, messages, jobs and other entities in the simulation. The HostArbiter manages the communication between the switch and the host, connected to it. All other switch components are related to the switch architecture, which is described in section 3.

6. Simulation Results

The Simulations were performed with three different packet sizes: 32 flits, 64 flits, 128 flits and three different traffic patterns (packet distributions): uniform, normal and exponential. In order to test each combination of packet size and traffic pattern, the simulation has to be performed 9 times. Each run of the simulation lasts for 2 hours and there are 20000 packets transmitted. The following charts show the mean latency for the packets, sent by the particular host (from the x axis). The hosts are numbered from 0 to 254. 0-240 are the hosts from clos 0 to clos 29. The remaining (241-253) belong to the root clos.

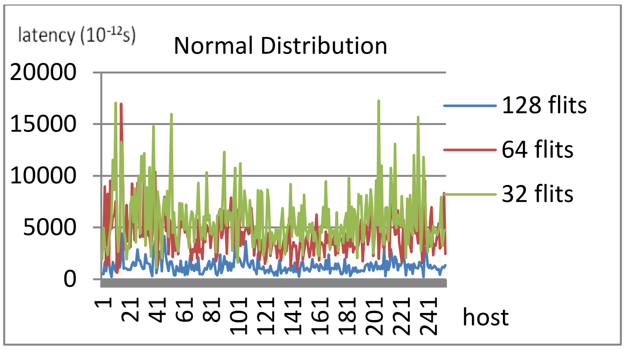

Fig. 9 Latency for normal packet distribution

There is latency increase for hosts 20-40 and 200-230. The reason for this behavior is that there are a lot of packets in the network, which are waiting on the wormhole principle, which leads to latency increase. At the start the network is being saturated with packets, and when there are too many packets, they need to wait and this leads to latency overhead.

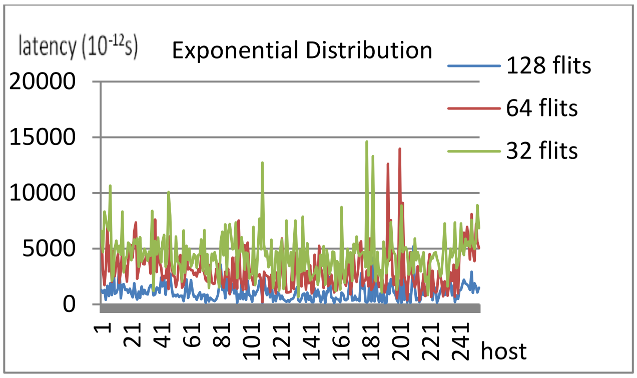

Fig. 10 Latency for exponential packet distribution

The last hosts (180-200 and 220-253) have increased packet delay, most visible for higher packet sizes. It is caused by the probability density of the distribution, which means that there are more packets, generated from the hosts with higher indexes.

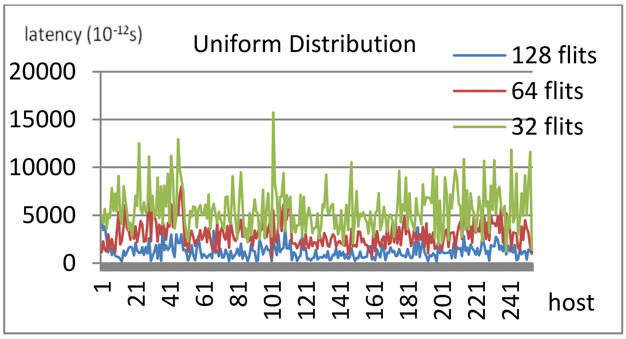

Fig. 11 Latency for uniform packet distribution

There are visible differences in the latency. However, in the 32 flits case, the results for all distributions are most consistent. Latency increase is observed at the hosts near 101. The reason for this is the peak value of the uniform distribution.

In order to show the advantages of the proposed Grand-Clos architectural design, the simulations are compared with a regular Fat Tree network under the same conditions (packet size, traffic pattern and total packets transmitted).

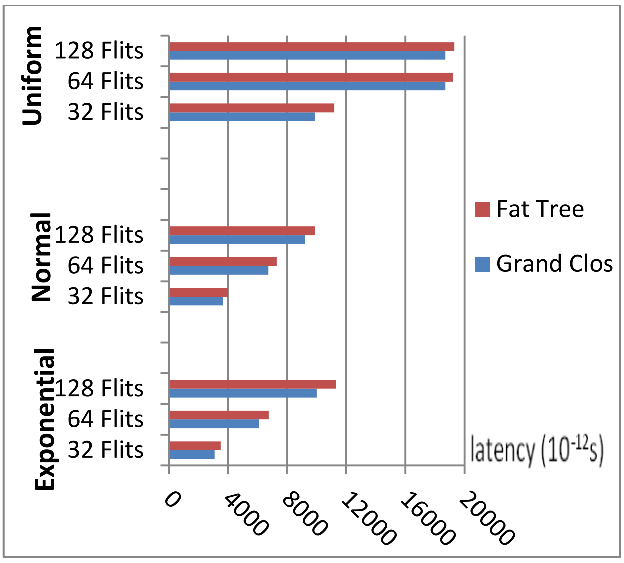

Fig. 12 OMNeT++ module hierarchy

The results from all simulations, provided in Fig.12, show that our Grand Clos Network achieves around 15% lower latency in all simulations. This performance increase explains the usage of the horizontal links in the proposed new network design of Grand Clos topology.

7. Future Work

We intend to perform more similar simulations in order to compare the proposed new Grand Clos topology with the topology used in BlueGene/P, which is Fat Tree/3D Torus. This will allow us to evaluate the behavior and to compare the performance of every of these topology designs.

Acknowledgment. The results reported in this paper are part of a research project DCVP 02/1 - Center of Excellence “Supercomputer Applications” – SuperCA++, financed by the National Science Fund, Bulgarian Ministry of Education and Science.

References

1. Dally W. J., Towels B., *Principles and practices of Interconnection Networks*, Morgan Kaufmann, ISBN-13: 978-0122007514, 2004

2. James Milano Gary L. Mullen-Schultz, Gary Lakner, *BlueGene-red book: Blue Gene/L: Hardware Overview and Planning*

3. P. Borovska. *Computer systems*. Sofia; Bulgaria: Ciela, ISBN 954-649-633-2 (in Bulgarian), 2009.

4. Duato, J., Yalamanchili, S., Lionel M., *Interconnection networks: An engineering approach*, Morgan Kaufmann Publishers, ISBN 1-55860-852-4, 2002.

5. http://omnetpp.org/doc

6. Pl. Borovska, O. Nakov , D. Ivanova, K. Ivanov, G. Georgiev, *Communication Performance Evaluation and Analysis of a Mesh System Area Network for High Performance Computers*. 12-th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligence Systems (MAMECTIS’10), Kantaoui, Sousse, Tunisia, May 3-6, 2010, ISBN: 978-960-474-188-5, pp. 217-222.

7. P. Borovska, O. Nakov, D. Ivanova, A. Ruzhekov, Halil Mohamed, *A Comparative Analysis of Next Generation High-End Switch Architectures*. Fifth International Conference "Computer Science", Bulgaria, Proceeding, pp. 7-12, 5-6 November 2009

8. P. Borovska, D. Ivanova, K. Ivanov, G. Georgiev, *Generalized Simulation Model of a Switch for High-Speed Interconnection Networks*, Sixth International Scientific Conference Computer Science’2011, Ohrid, Macedonia, pp. 17-22, 01 - 03 September 2011

9. Plamenka Borovska, Desislava Ivanova, Venelina Ianakieva, Vladislav Mitov, Halil Alkaf, *Comparative Analysis of Communication Performance Evaluation for Butterfly Bidirectional Multistage Interconnection Network Topology with Routing Table and Destination Tag Routing*, Sixth International Scientific Conference Computer Science’2011, Ohrid, Macedonia, pp. 29-34, 01 - 03 September 2011