American Journal of Networks and Communications
2014; 3(3): 29-38
Published online July 30, 2014 (http://www.sciencepublishinggroup.com/j/ajnc)
doi: 10.11648/j.ajnc.20140303.11
ISSN: 2326-893X (Print); ISSN: 2326-8964 (Online)

Analysis of effectiveness of communication overheads in the parallel computing system using stochastic colored petri nets

Nguyen Minh Quy[1], Huynh Quyet Thang[2], Ho Khanh Lam[1]

[1]Faculty of Information Technology, Hung Yen University of Technology and Education, Hung Yen, Vietnam
[2]School of Information Communication and Technology, Hanoi University of Science and Technology, Ha Noi, Vietnam

Email address:
quyutehy@gmail.com (N. M. Quy), thanghq@soict.hust.edu.vn (H. Q. Thang), lamhokhanh@gmail.com (H. K. Lam)

To cite this article:
Nguyen Minh Quy, Huynh Quyet Thang, Ho Khanh Lam. Analysis of Effectiveness of Communication Overheads in the Parallel Computing System Using Stochastic Colored Petri Nets. *American Journal of Networks and Communications.* Vol. 3, No. 3, 2014, pp. 29-38. doi: 10.11648/j.ajnc.20140303.11

Abstract: In architectures of parallel computing system, which has a large number of processing nodes, communication overhead is an important metric to evaluate and minimize by improving computation speedup solutions. In this paper, we propose using Stochastic Colored Petri Net to give models of parallel computing multi-processing systems for analyzing and evaluating effectiveness of communication overheads to system performance.

Keywords: SCPN, Parallel Computing System, Communication Overhead, Interconnecting Network

1. Introduction

We proposed the extension of the speedup formula of the Amdahl's law by including the communication overhead, and using the Closed Product Form Queueing Network (CPFQN) to analyze and evaluate effectiveness of the communication overhead (with changes of the data size, number of processing nodes and interconnecting network structures of processing nodes) on the speedup and performance of parallel computing multi-processing systems [1]. In this paper, we analyze and evaluate effectiveness of the communication overhead to performance of parallel computing system; another method is presented using Stochastic Colored Petri Nets (SCPN), in which transition times have an exponential distribution.

The Colored Petri Net (CPN) is a graphical oriented language for design, specification, simulation and verification of systems. It is in particular well-suited for systems in which synchronization, communication and resource sharing are important [2]. The CPN is a combination of Petri nets and programming language in which control structures, synchronization, communication and resource sharing are described by CPN. Data and data processing are described by functional programming language. Each CPN could be transformed to an equivalent PN and vice versa, i.e. if CPN has scalar data type (e.g. integers, strings and real numbers) then equivalent PN could be unlimited.

A CPN is defined as follow [3]:

$$CPN = (\sum, P, T, A, N, C, G, E, I)$$

Where:

$\sum$ - Finite set of non-empty types, also called finite and non-empty colour sets.

P - Finite set of places.

T - Finite set of transitions.

A - Set of arcs such that $P \cap T = P \cap A = T \cap A = \phi$.

N- Node function, defined from a into "colour on arcs" $P\,x\,T \cup T\,x\,P$, is written by $N : A \to P\,x\,T \cup T\,x\,P$.

$C : P \to \sum$ - colour function assigned to each place $p \in P$ one colour set $C(p) \in \sum$.

$G : T \to EXP$ - Guard function assigned to each transition $t \in T$ one guard expression as "Boolean function of probability" and is written as:

$\forall t \in T : [Type(g(t)) = Boolean \wedge Type(Var(g(t))) \subseteq \sum]$ w here $Type(Vars)$ is set of types $\{Type(v) \mid v \in Vars\}$, $Vars$ is set of variations, $Vars(G(t))$ are variations in $g(t)$.

$E : F \to EXP$- An arc expression function assigned to each arc $a \in A$ one expression including multi-set

$C(p)_{MS}$:

$\forall a \in A: [Type(E(a)) = C(p(a))_{MS} \wedge Type(\text{var}(E(a))) \subseteq \sum]$

where, $p(a)$ is the place of $N(a)$ and $C(p)_{MS}$ denotes the set of all multi-sets over C

$I: P \to EXP$ – An initialization function assigned to each place $p \in P$ one expression including multi-set $C(p)_{MS}$:

$$\forall p \in P: [Type(I(p)) = C(p)_{MS} \wedge Var(I(p)) = \phi]$$

Graphical representation of CPN consists of ellipses and circles denoted places which describe states of the system (buffers). The rectangles designate transitions which describe actions (processes). The arrows designate arcs. Expressions of arcs describe variable status of CPN when transitions occur. Each place contains a set of marker called tokens. In the CPN, each of these tokens carries a data value, which belongs to a given type and could be different from other one, while all tokens are the same in common Petri nets. CPNs create condensation network models by using definitions of colours. In the SCPN, there are immediate transitions and transitions associated with delay times of firing, that are called timed transitions.

Nowadays, there are some researches that used CPN to model, analyze and evaluate many systems in different fields, especially on information technology and communication. Recently, CPN was used to model parallel systems in [4], [5] and [6]. However, communication overhead problem of modern parallel computing systems is slightly assessed. Kotov et al. [7] presented a mesh network structure and used Design/CPN tool to analyze and evaluate. In this study, we present the cluster structure of processing nodes to analyze and evaluate since the cluster structures are typical in architectures of parallel computing systems.

2. SCPN Model of Parallel Computing Multi-Processing System

There are some versions for modeling tool of CPN such as CPN tools (Aarhus University, Denmark) and TimeNet (Armin Zimmermann and Knoke, Technical University of Berlin, Germany). We used TimeNet 4.0.2 tool to model the parallel computing multi-processing system. We chose the cluster interconnection parallel system with commonly used configurations 2Dtorus, in which each processing node connects to four adjacent nodes (degree of node is four). The present model consists of five processing nodes: node P0 connects to four adjacent nodes P1, P2, P3 and P4. The structure and technology of each node Pi (i=0,1,2,3 and 4) are identical as shown in Figure 1.

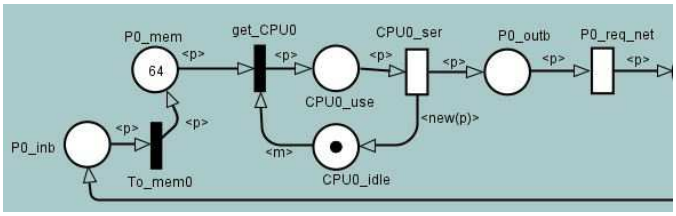


Figure 1. *SCPN of processor*

Table 1. *List of places in processor*

Name	Token type	Token list	Queue	Meaning
P0_mem0	int	8, 16, 32, 64	Random	Processors: number of initial tokens representing number of data packet with content =1 for P0 (expression on output arc p = = 1)
CPU0_use	int		Random	CPU processes data packet
CPU0_idle	int	1	Random	Idle CPU, initial state is established by one token
P0_outb	int		Random	Output buffer of processor, initial state is empty
P0_inb	int		Random	Input bufer of processor, initial state is empty

Table 2. *Token list of processor has identical content for all data packets (P0: content = 1, P1: content =2, P2: content =3, P3: content = 4, and P4: content = 5).*

Tokenlist	
Number	**Content**
1	1.0
2	1.0
3	1.0
4	1.0
5	1.0

Tokenlist	
Number	**Content**
6	1.0
7	1.0
8	1.0
9	1.0
10	1.0
11	1.0
12	1.0
13	1.0

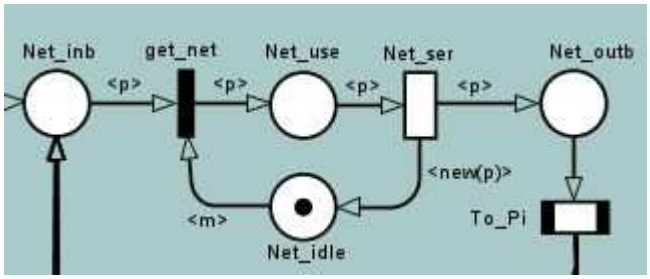
Table 3. *Transitions having firing delay of processor*

Name	timeFunction	localGuard	Meaning
CPU0_ser	EXP (CPU_service_time)	P	Time for completely processing data packet of CPU (CPU_service_time)
P0_req_net	EXP (net_access_time)	P	Time for assessing communication request from processor to interconnect network

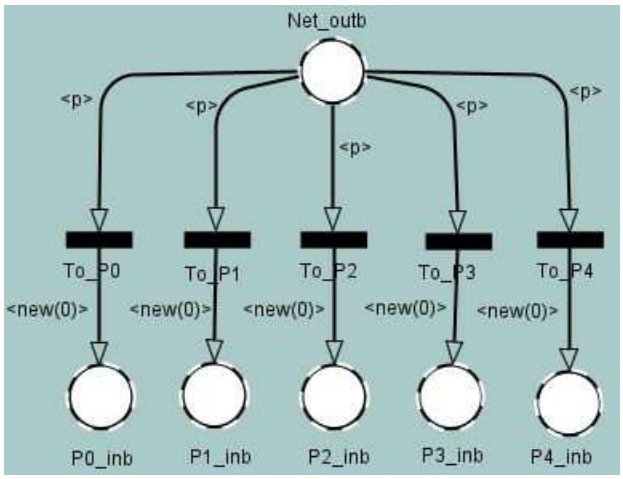
Table 4. *Immediate transitions of processor (time delay = 0)*

Name	Priority	weight	localGuard	Meaning
get_CPU0	1	1.0E0	p	Immediately fires as having data packet in memory
To_mem0	1	1.0E0	p	Immediately fires as communication from other processor across interconnect network

Model of cluster interconnect network is presented in *Figure 2*.



(a) SCPN of interconnect



(b) Sub-SCPN of Interconnect

Figure 2. *SCPN of interconnecting network*

Table 5. *List of places of processor*

Name	Token type	Tokenlist	Queue	Meaning
Net_inb	int		Random	Input buffer of interconnect
Net_use	int		Random	Interconnect processes data packet
Net_idle	int	1	Random	Idle interconnect, initial state is established by one token
Net_outb	int		Random	Output buffer of Interconnectr, initial state is empty

Table 6. *Transitions having firing delay*

Name	timeFunction	localGuard	Meaning
Net_ser	EXP(net_server_time)	p	Communication time across interconnect (net_server_time)

Table 7. *Immediate transition of interconnect (time delay = 0)*

Name	Priority	weight	localGuard	Meaning
get_net	1	1.0E0	p	fires as having request of communication interconnect
To_P0	1	1.0E0	p	fires as having token at output buffer of interconnect: transition of data packet to processors

In the parallel computation, a processor node locally implements some tasks such as computation and sends of computing results to other processors that could be adjacent or remote depending on parallel algorithm of applications. Therefore, the data packets from a processor will be sent to other processors without return itself. The communication overhead could be smallest if sending to adjacent processors or average value based on H- average routing

distance (average hop count) of each configuration. For example, with 2Dtorus H = √n/2, *n* is node number of processors. Node P0 only transfers data packet across network to P1, P2, P3 and P4. Figure 3 presents SCPN of parallel computing system with 2Dtorus cluster interconnect.

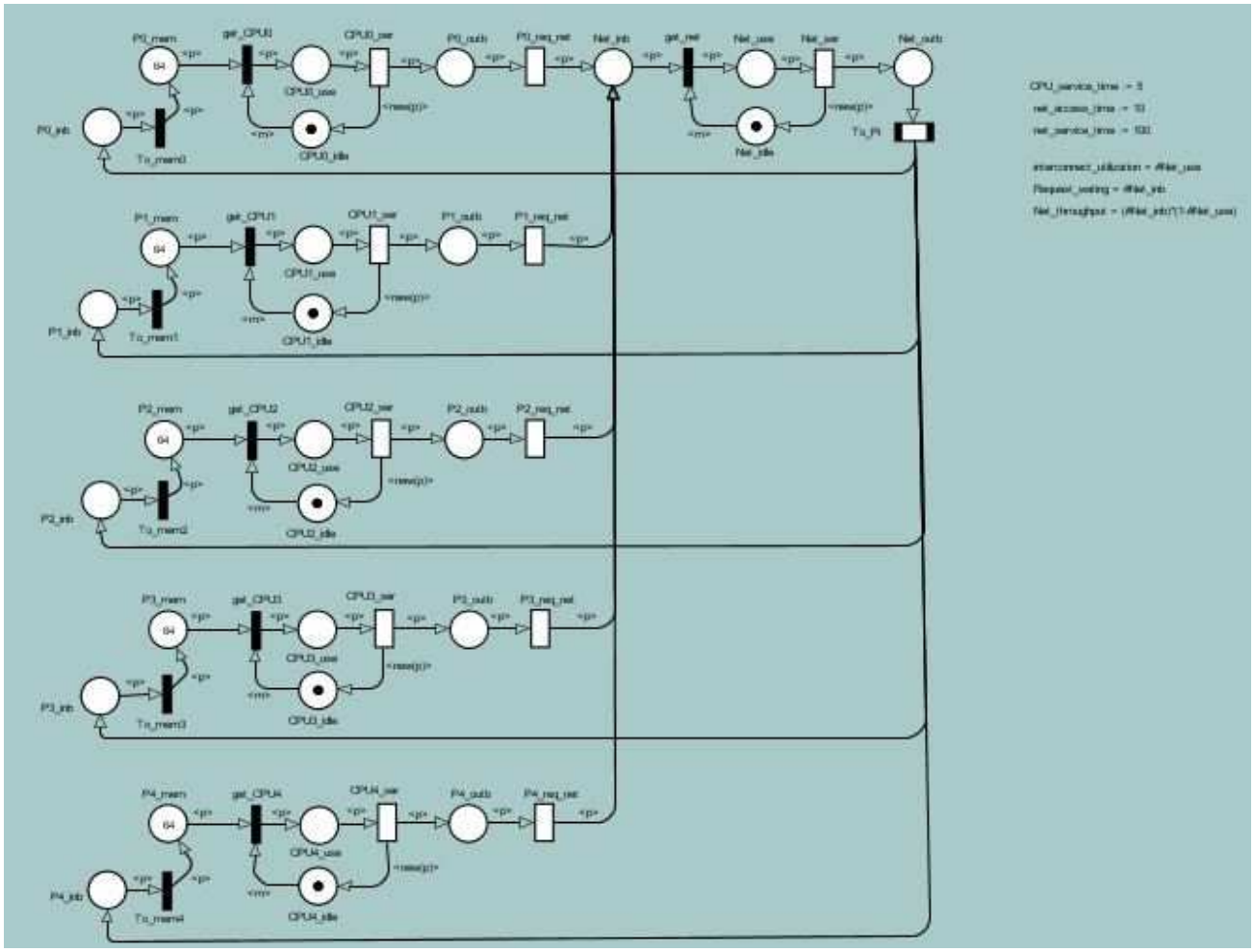


Figure 3. *SCPN of the cluster 2Dtorus interconnection parallel computing system*

In this model, initial states of places P0_mem (P1_mem, P2_mem, P3_mem and P4_mem) of respective processor nodes (P0, P1, P2, P3 and P4) have been established with number of tokens 8, 16, 32 and 64 (preset numbers of packets) for simulation scenarios. The firing of immediate transitions get_CPU0 (get_CPU1, get_CPU2, get_CPU3 and get_CPU4) presents that the data packets of an application are sent to the CPU of the processor node for processing. Places CPU0_use (CPU1_use, CPU2_use, CPU3_use and CPU4_use) have tokens present processing states of respective CPUs. Processing delay time, CPU_service_time, of processors is exponential distributed and is defined by value EXP (CPU_service_time). The data packets, which need to send to processors across interconnect, are transferred to output buffer by the processor which present by places P0_outb (P1_outb, P2_outb, P3_outb and P4_outb). From output buffers, processors sent to interconnect and are defined by interconnect access delay, net_access_time, that is exponential distribution, defined by value EXP(net_access_time). The communication requests from processors are input to input buffer of interconnect represented as place Net_inb. Normally, interconnect is connected to buffers by switches or routers to process information routing. When Net_inb has communication request, get_net transition is activated and interconnect communicates data packets (information routing, transition of data packets). Status of communicating execution of interconnect is represented by Net_use position. Net_ser transition has communication delay (or activate delay), net_service_time, that is an exponential distribution, EXP(net_service_time). After communication delay across interconnect, the data packets are transferred to output buffer of interconnect and To_P0 transitions (To_P1, To_P2, To_P3 and To_P4) (in Figure 2b) are immediately activated to transfer the data packets to input buffers of target processor nodes, i.e. P0_inb positions (P1_inb, P2_inb, P3_inb and P4_inb), respectively. At processor nodes, occurrence of token at P0_inb position (P1_inb, P2_inb, P3_inb and P4_inb) makes To_mem0 transitions

(To_mem1, To_mem2, To_mem3 and To_mem4) immediately activated. The data packets are transferred to respective memories. Communication procedure among processor nodes in the parallel system finishes with a data packet. In the present study, data packet size is not important since the number of data packet transferred at a setting period is effective to the communication overhead. Moreover, the number of processing node, n, interconnect configuration and data packet size are effective to parameters: net_access_time, net_service_time. These parameters will be changed in simulation scenarios.

The functions, <p>, at arcs ensure that tokens are transferred to respective tokens in processor positions with the same content. The functions, <m>, at arc ensure new free status for intranet and interconnect networks. The functions, new (0), determine completely communicating implementation of requests from processors. Performance parameters of this model are determined and presented in Table 8.

Table 8. *Performance parameters*

Parameters	Value	Meaning
Interconnect_utilization	#Net_use	Utilization of interconnect network is number of token (number of requests of served processors) at Net_use position. Utilization of interconnect is high then communication overhead across interconnect is too high and throughput is decreased.
Request_waiting	#Net_inb	Number of token having at Net_inb position presents communication requests (number of data packets) from processor waiting communication acquisition of Interconnect.
Net_throughput	(#Net_inb)*(1-#Net_use)	Throughput of interconnect

3. Results and Discussion

For simulation, there are two following scenarios:

Scenario 1: set CPU_service_time:=5, net_access_time:=10, net_service_time:=100 and processors have 8, 16, 32 and 64 data packets (number of tokens). Determine performance parameters at maximum, mean and minimum values for Interconnect_utilization, Request_waiting and Net_throughput. Results simulated on TimeNet4.0.2 are represented in Figs. 4, 6, 8 and 10 with period of second. The ordinate axis presents performance parameters and the horizontal axis presents modeling time in second.

Scenario 2: set CPU_service_time:=5, net_access_time:=10, net_service_time:=300 and processors have 8, 16, 32 and 64 data packets (number of tokens). Determine performance parameter at maximum, mean and minimum values for Interconnect_utilization, Request_waiting and Net_throughput. Results simulated on TimeNet4.0.2 are represented in Figs. 5, 7, 9 and 11 with period of second. The ordinate axis presents performance parameters and the horizontal axis presents modeling time in second.

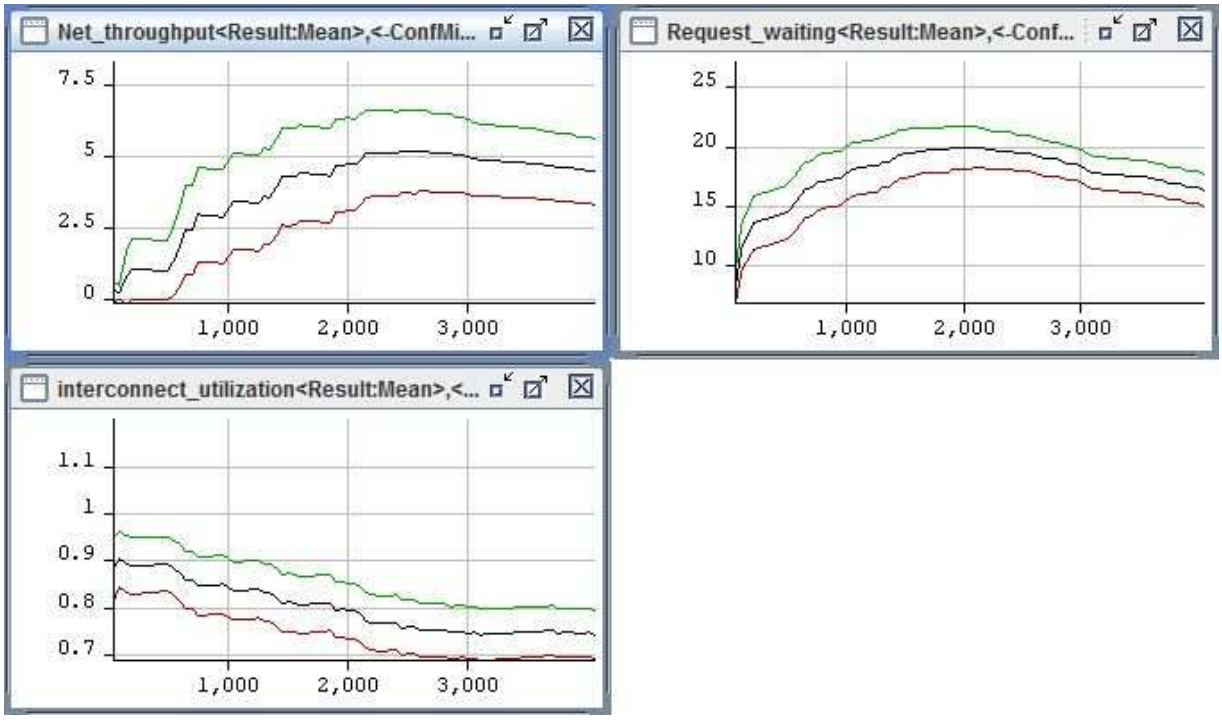


Figure 4. *Scenario 1 with 8 data packets, CPU_service_time=5, net_access_time=10 and*

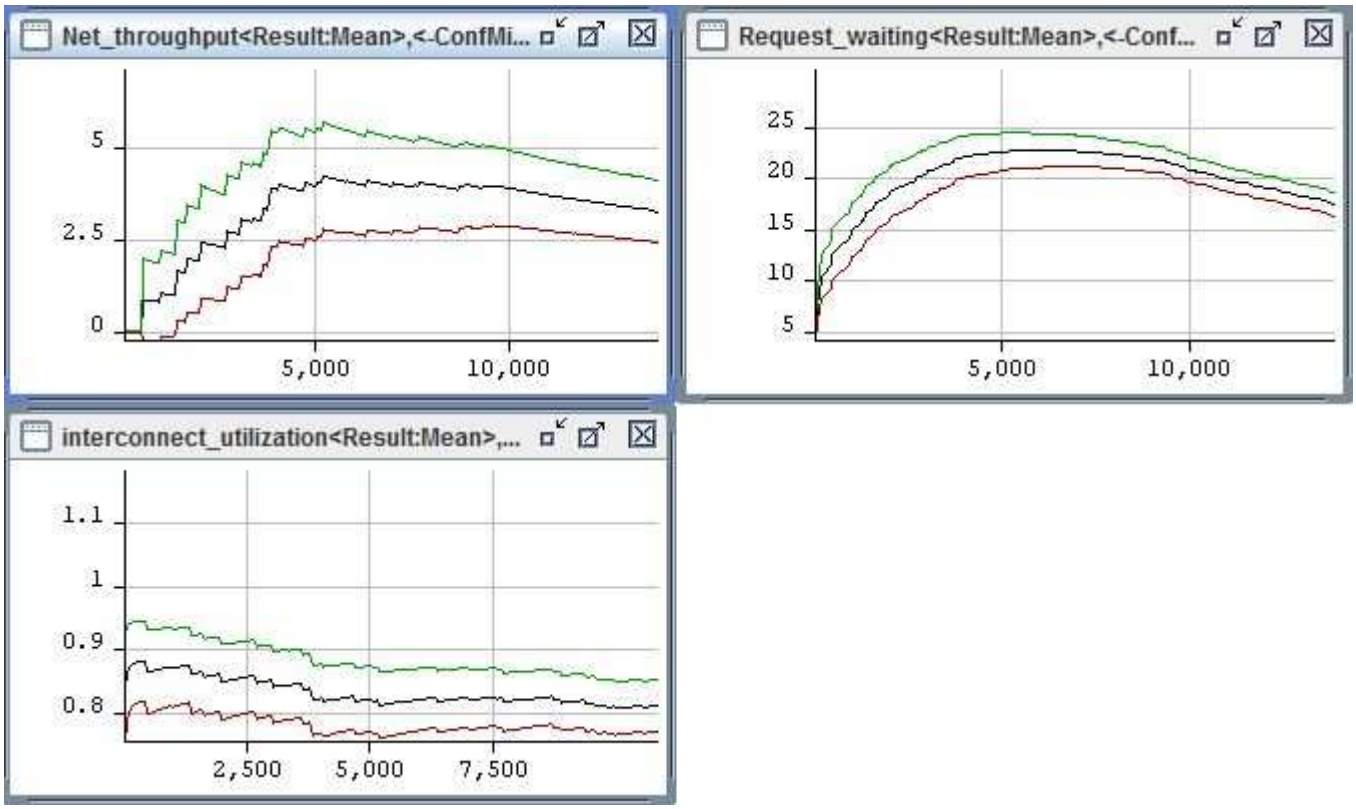


Figure 5. *Scenario 2 with 8 data packets, CPU_service_time=5, net_access_time=10 and net_service_time=300.*

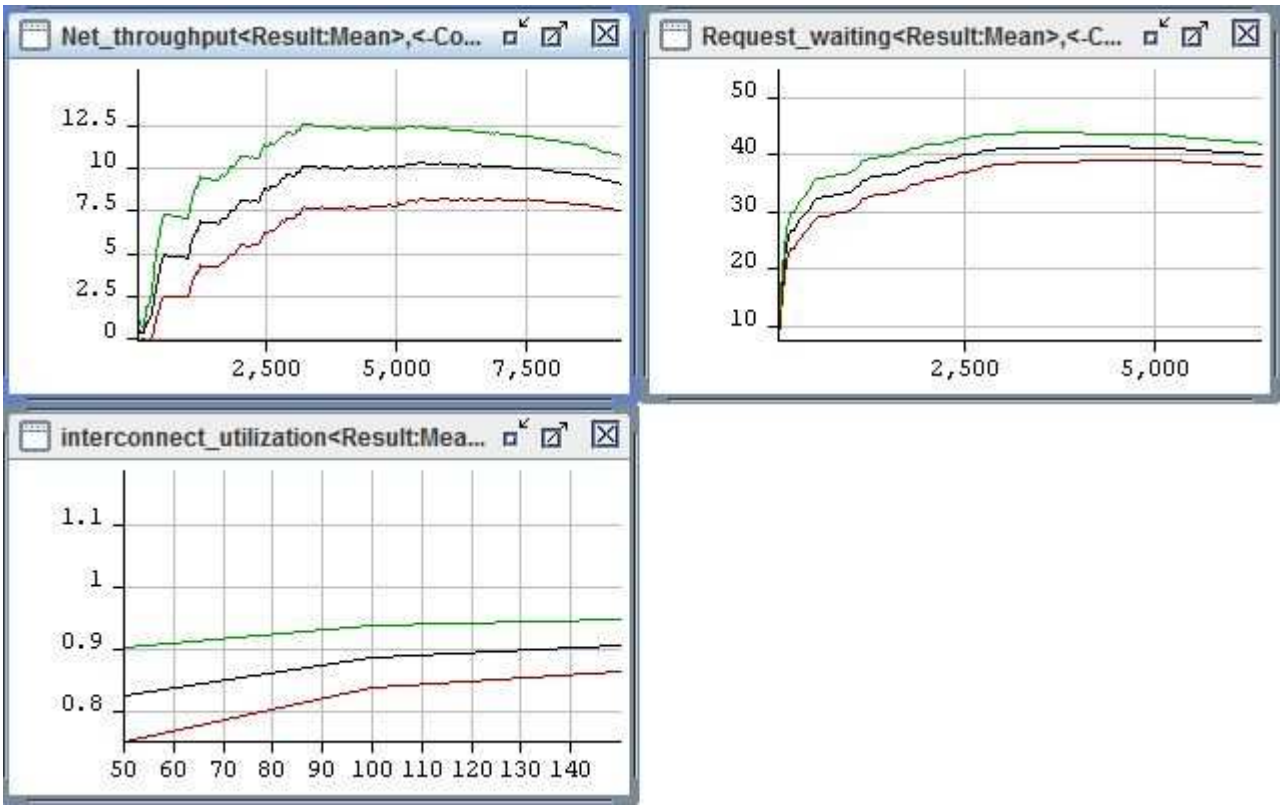


Figure 6. *Scenario 1 with 16 data packets, CPU_service_time=5 and net_access_time=10, net_service_time=100.*

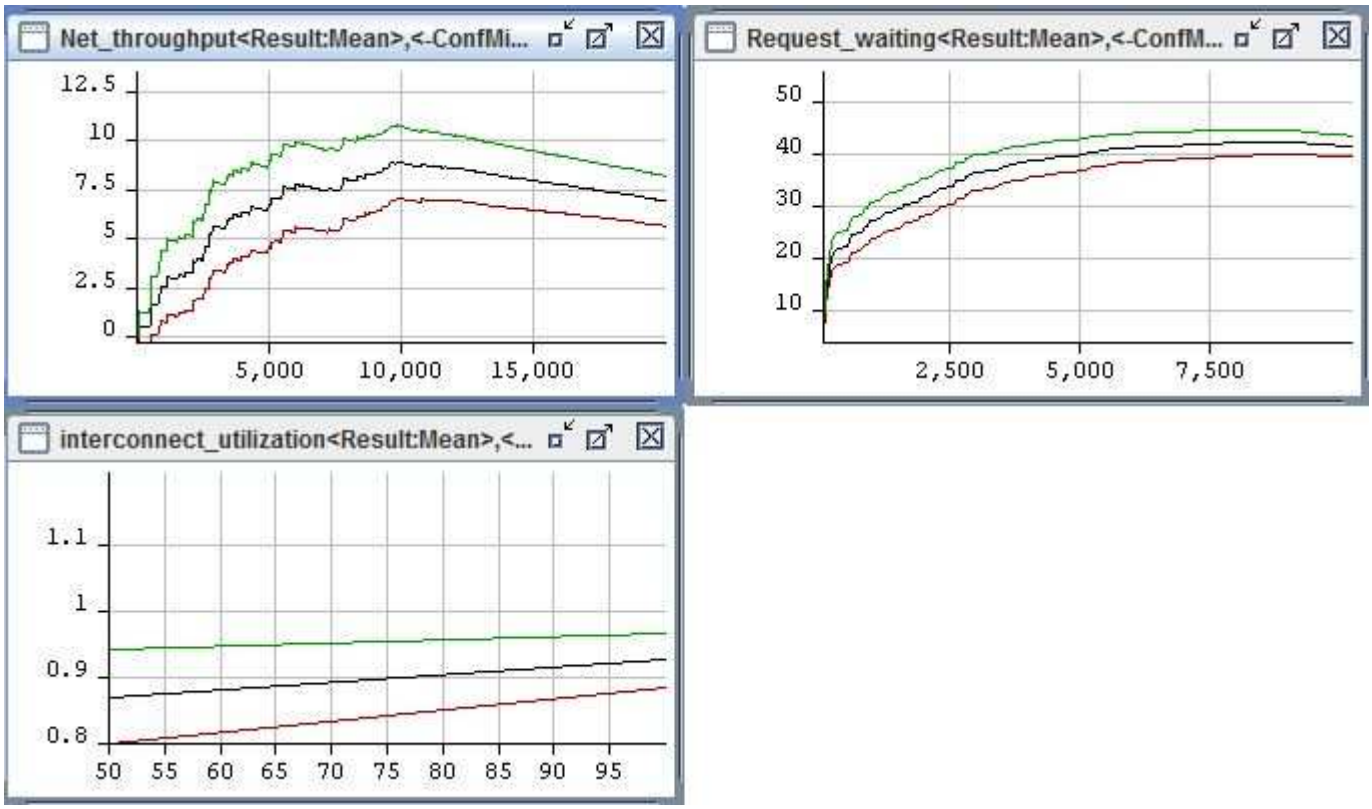


Figure 7. *Scenario 2 with 16 data packets, CPU_service_time=5, net_access_time=10 and net_service_time=300.*

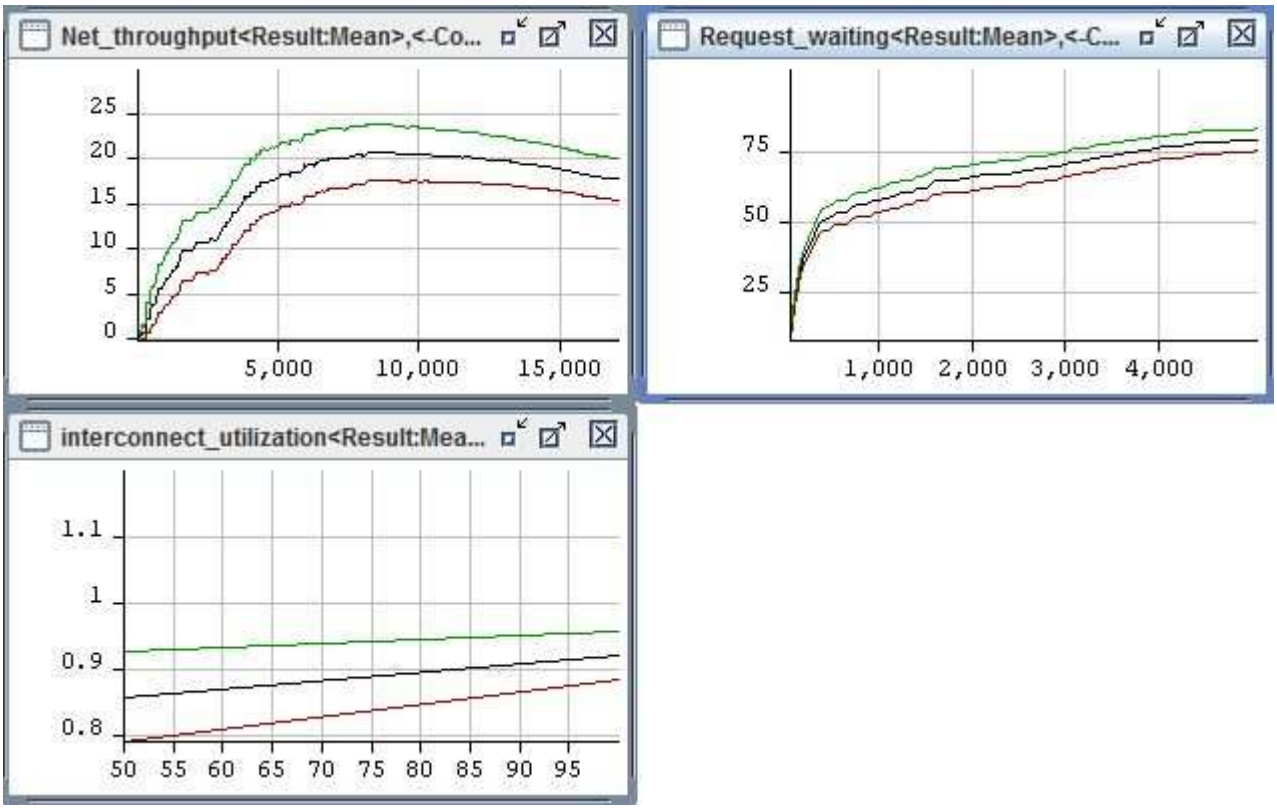


Figure 8. *Scenario 1 with 32 data packets, CPU_service_time=5, net_access_time=10 and net_service_time=100.*

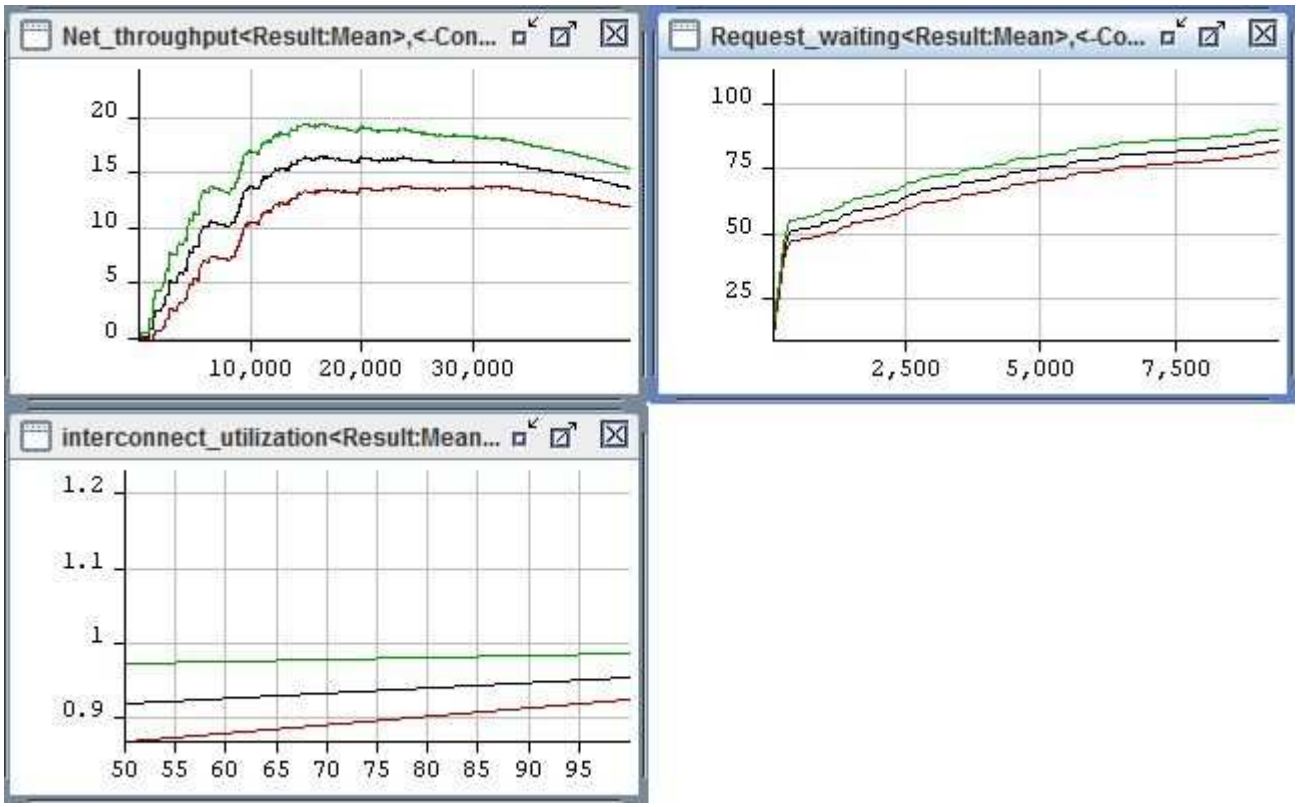


Figure 9. *Scenario 2 with 32 data packets, CPU_service_time=5, net_access_time=10 and net_service_time=300.*

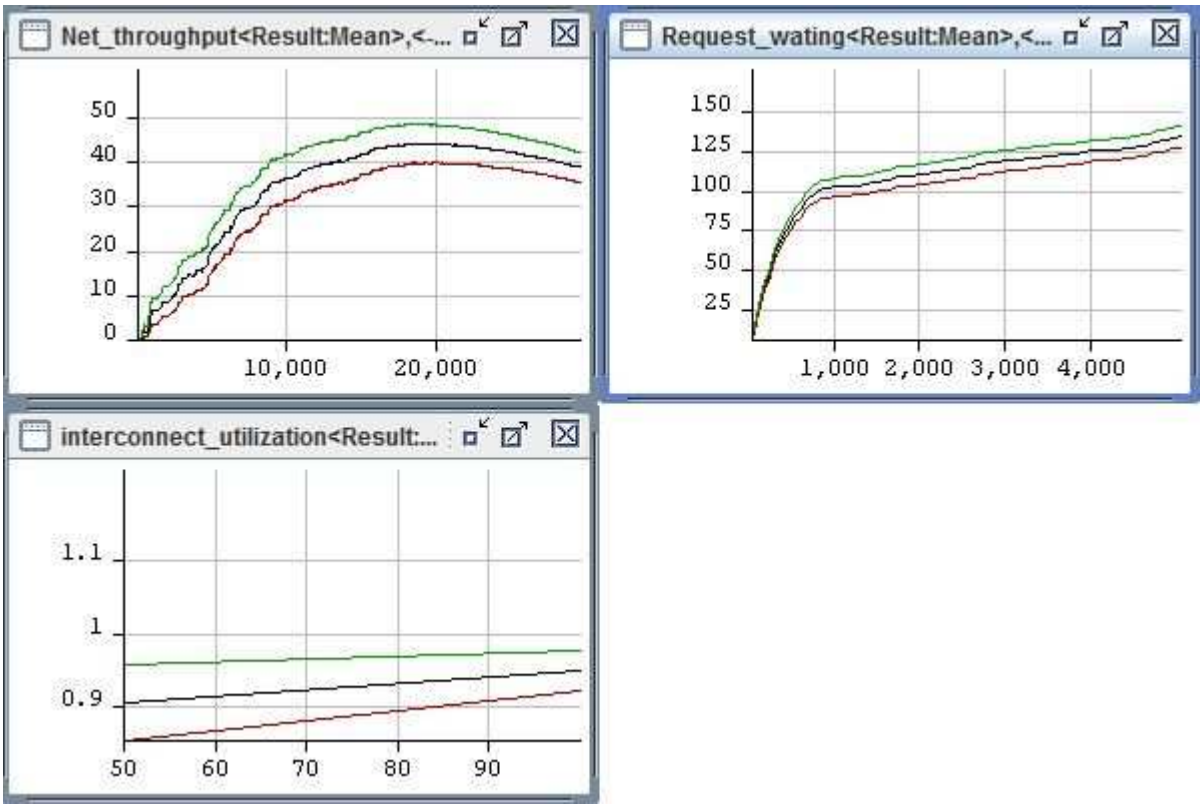


Figure 10. *Scenario 1 with 64 data packets, CPU_service_time=5, net_access_time=10 and net_service_time=100.*

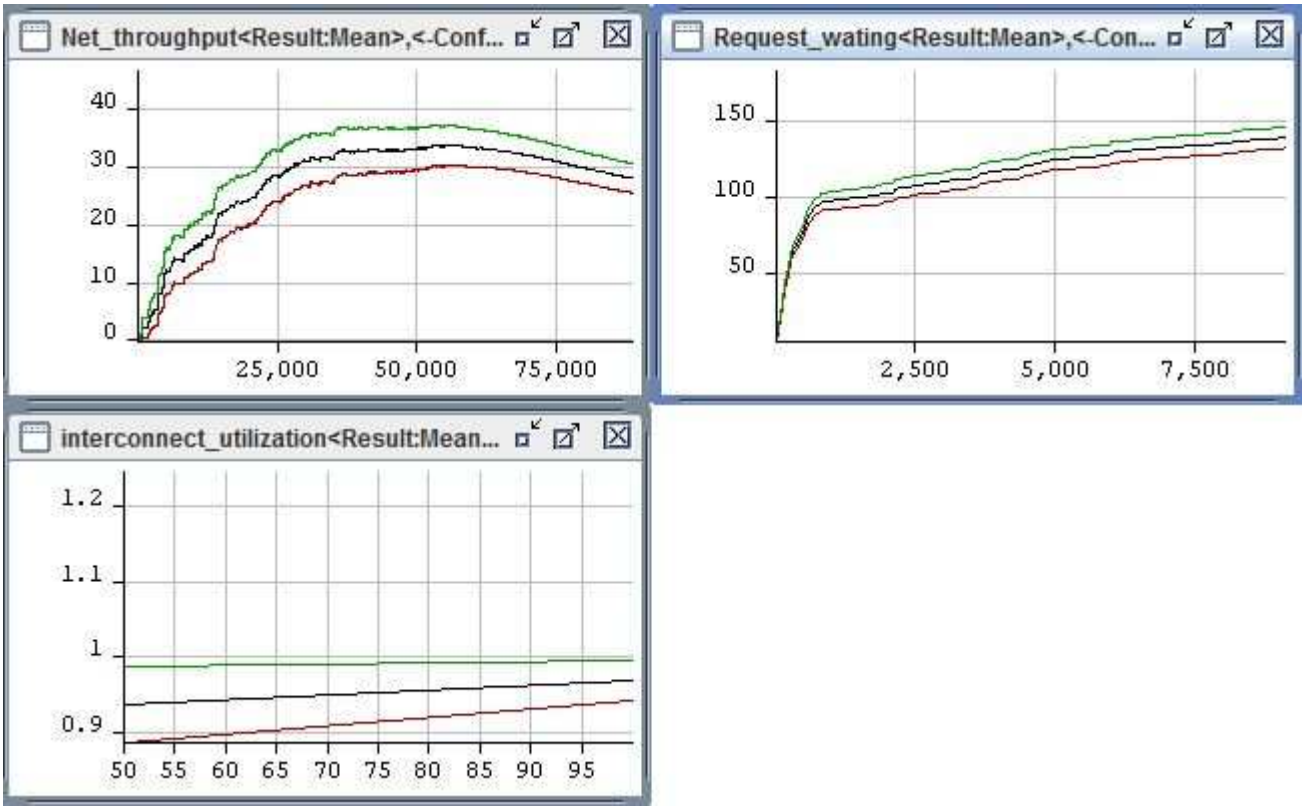


Figure 11. *Scenario 2 with 64 data packets, CPU_service_time=5, net_access_time=10, net_service_time=300.*

From simulation results of SCPN, with a large number of data packets, the utilization is large and numbers of data packets that must be wait are too large. The throughput is decreased as keeping mean values of CPU_service_time, net_access_time and net_service_time. When the communication delay time of interconnect (net_service_time) increases, performance of interconnect is decreased. By using SCPN, models of SCPN for any network interconnect structures could be established and their performance parameters could be also assessed. In the present study, a model of SCPN for 2Dtorus structure is presented to illustrate SCPN using in assessing performance of multi-processing parallel computing systems.

Our purpose is putting out the method using SCPN for assessing performance of multi-processing parallel computing systems depended on communication overhead of interconnect network of processing nodes, in which topologies of interconnect network and number of data packets transmitted between processing nodes are considered.

4. Conclusions

In the present study, we have successfully given models of parallel computing multi-processing systems for analyzing and evaluate effectiveness of communication overheads to system performance using Stochastic Colored Petri Net. The cluster structure of processing nodes have been also analysed and evaluated. A model of SCPN for 2Dtorus structure was used to illustrate SCPN using in assessing performance of multi-processing parallel computing systems. With a large number of data packets, the utilization is large and numbers of data packets that must be waiting are too large. The throughput is decreased as keeping mean values of CPU_service_time, net_access_time and net_service_time. When the communication delay time of interconnect increases, performance of interconnect is decreased. We could have a more complicated SCPN for a multi-processing parallel computing system if the MPI protocol commonly used in parallel architectures is considered, and select some types of applications for simulation as matrix multiplication, image processing.

References

[1] Nguyen Minh Quy, Ho Khanh Lam, Huynh Quyet Thang, "Analysis of Effectiveness of Communication Overheads in the Parallel Computing System Using the Closed Product Form Queuing Network", RIVF-2013: The 10th IEEE RIVF International Conference on Computing and Communication Technologies, Hanoi, Vietnam, 10-13 November 2013, pp. 131-134.

[2] K. Jensen, "An Introduction to the Theoretical Aspects of Coloured Petri Nets". Lecture Notes in Computer Science vol. 803, Springer-Verlag 1994,230-272.

[3] K. Jensen,"Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use". Monographs in Theoretical Computer Science, Springer-Verlag, 2nd corrected printing 1997, ISBN: 3-540-58276-2.

[4] Homayun Motameni, Zohre Ramezani and Zahra Usefi, "Modeling and Simulation of Parallelism by Colored Petri Nets". World Applied Sciences Journal 19 (5): 710-713, 2012. ISSN 1818-4952; © IDOSI Publications, 2012.

[5] Stanislav Böhm, Marek Běhálek, "Usage of petri nets for high performance computing**".** FHPC '12 Proceedings of the 1st ACM SIGPLAN workshop on Functional high-performance computing. Pages 37-48. ACM New York, NY, USA ©2012. ISBN: 978-1-4503-1577-7.

[6] Bin Cheng, Weiqin Tong, and Xingang Wang, "Hybrid Performance modeling and analyzing of parallel systems". International Journal of numerical analysis and modeling, Volume 9, Number 2, Pages 232-246, © 2012 Institute for Scientific Computing and Information.

[7] Gianfranco Ciardo, Ludmila Cherkasova, Vadim Kotov, and Tomas Rokicki, "Modeling A Scalable High-Speed Interconnect with Stochastic Petri Nets". Department of Computer Science College of William and Mary USA, Hewlett-Parkard Labs. 1994.