Analysis of Adaptive Wormhole-Routed Torus Networks with IPP Input Traffic

Geyong Min
Department of Computer Science
University of Strathclyde
Glasgow, G1 1XH, U.K.
Email: geyong@cs.strath.ac.uk

John Ferguson
Department of Computer Science
University of Strathclyde
Glasgow, G1 1XH, U.K.
Email: jf@cs.strath.ac.uk

Mohamed Ould-Khaoua
Department of Computing Science
University of Glasgow
Glasgow, G12 8QQ, U.K.
Email: mohamed@dcs.gla.ac.uk

ABSTRACT

Recent work has shown that there exists burstiness characteristic (i.e. time-varying arrival rates) in multimedia traffic. It is important to understand the performance of interconnection networks in the presence of this kind of traffic. However, simulation-based approaches may be costly and time-consuming. This paper proposes a new analytical model for adaptive wormhole-routed torus network with interrupted Poisson process (IPP) input traffic. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments.

Keywords

Multicomputers, Interconnection Networks, Message Latency, Performance Modelling, Queueing Theory.

1. INTRODUCTION

Parallel and distributed systems (or multicomputers) are commonly accepted as good candidates for large-scale multimedia servers as they meet the high computation and communication requirements of multimedia applications. The efficiency of multicomputers is critically dependent on the performance of its interconnection network, which is determined by the topology, switching method, routing algorithm and traffic model [5].

Torus networks have been employed extensively in the latest generation of multicomputers [5]. Recently, Duato [4] proposed an adaptive routing algorithm to improve the performance of interconnection networks with wormhole switching. The algorithm divides the virtual channels into two classes: *a* and *b*. At each routing step a message can adaptively visit any available virtual channel from class *a*. If all the virtual channels belonging to class *a* are busy it crosses a virtual channel from class *b* using dimension-ordered routing.

Recent studies have revealed that traffic generated by multimedia application exhibits a high degree of burstiness, which can significantly affect queueing performance [6], [7]. However, all existing performance analyses of interconnection networks have assumed the traffic follows the Poisson arrival process [2], [3], [10], [11]. Using realistic traffic models is a critical step towards understanding the important factors that affect the quality-of-service of multimedia applications in multicomputers. The interrupted Poisson process (IPP), which is a special case of the Markov-modulated Poisson process (MMPP) [6], has been extensively used to model multimedia traffic in a single ATM mutiplexer [7], [8], [12].

This paper proposes a new analytical model for adaptive wormhole-routed torus networks with IPP input traffic by extending the application of MMPP to the whole network. We first approximate the arrival process at network channels to a two-state MMPP and treat network channel as an MMPP/G/1 queueing system. Then we derive the mean message latency in the network. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments. The rest of the paper is organised as follows. Section 2 describes the analytical model. Section 3 validates it through simulation. Finally, Section 4 concludes this study.

2. THE ANALYTICAL MODEL

The 2-dimensional torus network contains $N = k^2$ nodes, arranged in two dimensions, with *k* nodes per dimension. Each node is connected to its neighboring nodes through 4 inputs and 4 output channels [5]. The model is based on the following assumptions [1], [2], [3], [10], [11].

a) Traffic generated by the source nodes is independent of each other, and follows an IPP.
b) Message destination nodes are uniformly distributed across the network nodes.
c) Message length is *m* flits, where *m* is a random variable. The Laplace-Stieltjes transform of *m* is given by $M^*(s)$. Each flit requires one cycle to cross from one router to the next.
d) The local queue in the source node has infinite capacity. Moreover, messages at the destination node are transferred to the local *processing element* as soon as they arrive at their destinations.
e) Each physical channel is divided into *V (V>2)* virtual channels. In Duato's routing algorithm [4], class *a*

SAC 2001, Las Vegas, NV
© 2001 ACM 1-58113-287-5/01/02…$5.00

contains $(V-2)$ virtual channels and class b contains two virtual channels.

The mean message latency is composed of the mean network latency, $\bar{t}$, that is the time to cross the network, and the mean waiting time seen by message in the source node, $\bar{w}_s$. However, to model the effect of virtual channels multiplexing, the mean message latency has to be scaled by a factor, $\bar{v}$, representing the average degree of virtual channels multiplexing, that takes place at a given physical channel. Therefore, we can write [2], [10], [11]

$$Latency = (\bar{t} + \bar{w}_s)\bar{v} \tag{1}$$

Under the uniform traffic pattern, the average numbers of hops that a message makes along a given dimension and across the network, $\bar{k}$ and d, are given by [1]

$$\bar{k} = \begin{cases} k/4 & \text{if } n \text{ is even} \\ (k - 1/k)/4 & \text{if } n \text{ is odd} \end{cases} \tag{2}$$

$$d = 2\bar{k} \tag{3}$$

The network latency consists of two parts: one is the delay due to the actual message transmission time, and another is due to the blocking time in the network. Let t be a random variable that denotes the network latency. t can be written as

$$t = m + d + b \tag{4}$$

where m is the message length, d is the mean message distance, b is the blocking time experienced by a message to cross the network. Adaptive routing allows a message to cross in any order those channels that bring it closer to its destination resulting in an equal and balanced traffic load on all channels. Therefore, a message sees the same waiting time across all channels. However, it sees a different probability of blocking at each hop as the number of alternative paths changes from one hop to another [11]. The blocking time, *b*, is given by

$$b = \sum_{i=0}^{d} P_{b_i} w_b \tag{5}$$

where w_b and P_{b_i} denote the waiting time and probability that a message is blocked after making *i* hops.

Since the Laplace-Stieltjes transform of the sum of two independent random variables is equal to the product of their transforms [9], the Laplace-Stieltjes transform of the network latency is given by

$$T^*(s) = \int_0^\infty e^{-sx} dT(x) = M^*(s) e^{-sd} B^*(s) \tag{6}$$

where $M^*(s)$ and $B^*(s)$ denote the Laplace-Stieltjes transforms of the message length and blocking time. In what follows, we will describe the calculation of the following quantities: $B^*(s)$, $\bar{t}$, $\bar{w}_s$ and $\bar{v}$.

2.1. Determination of Traffic Characteristic at Network Channels

The traffic generated by a source node is represented by an independent IPP, which is a special case of the two-state MMPP(2) [6]. The MMPP(2) is a doubly stochastic process with an arrival rate governed by a two-state continuous time Markov chain. In state i (i=1, 2), the arrival process follows a Poisson process with rate μ_i. The transition rate out of state 1 to 2 is σ_1, while the rate out of 2 to 1 is σ_2. In particular, the MMPP(2) with one of the arrival rates σ_1 or σ_2 equal to 0 is called an IPP. The MMPP(2) can be parameterised by the infinitesimal generator $\mathbf{Q}_s$ of the underlying Markov chain and the rate matrix Λ_s [6]

$$\mathbf{Q}_s = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix}, \ \Lambda_s = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \tag{7}$$

The mean arrival rate, λ_s, and the counting function $N_s(t)$, the number of arrivals in $(0,t]$, of the MMPP(2) play a major role in the subsequently described method to obtain the arrival process at network channels. We recollect from [7] the formulae for the mean, variance-to-mean ration, the third moment of $N_s(t)$ and the mean arrival rate as

$$E[N_s(t)] = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} t \tag{8}$$

$$\lambda_s = \frac{E[N_s(t)]}{t} = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} \tag{9}$$

$$\frac{Var[N_s(t)]}{E[N_s(t)]} = 1 + \frac{2(\mu_1 - \mu_2)^2 \sigma_1\sigma_2}{(\sigma_1 + \sigma_2)^2(\mu_1\sigma_2 + \mu_2\sigma_1)} - \frac{2\sigma_1\sigma_2(\mu_1 - \mu_2)^2(1 - e^{-(\sigma_1+\sigma_2)t})}{(\sigma_1 + \sigma_2)^3(\mu_1\sigma_2 + \mu_2\sigma_1)t} \tag{10}$$

$$\lim_{t\to\infty} \frac{Var[N_s(t)]}{E[N_s(t)]} = 1 + \frac{2(\mu_1 - \mu_2)^2 \sigma_1\sigma_2}{(\sigma_1 + \sigma_2)^2(\mu_1\sigma_2 + \mu_2\sigma_1)} \tag{11}$$

$$E[(N_s(t) - E[N_s(t)])^3] = g^{(3)}(1,t) - E[N_s(t)](E[N_s(t)] - 1)\left(\frac{3Var[N_s(t)]}{E[N_s(t)]} + E[N_s(t)] - 2\right)$$

$$g^{(3)}(1,t) = \frac{6}{\sigma_1 + \sigma_2}[\frac{A_{11}}{6}t^3 + \frac{A_{21}}{2}t^2 + A_{31}t + A_{12}te^{-(\sigma_1+\sigma_2)t} + A_{41}(1 - e^{-(\sigma_1+\sigma_2)t})] \tag{12}$$

Expressions for A_{ij} are given in [7] in terms of the four parameters μ_1, μ_2, σ_1 and σ_2.

Under the uniform traffic pattern, adaptive routing results in an equal and balanced traffic load on all network channels. The arrival process at network channels exhibit similar statistical behaviour. Each message travels, on average, *d* channels to reach its destination and each node has 4 output channels. As a result, the amount of traffic that arrives at each network channel, on average, is equal to the amount of traffic generated by n_s source nodes. n_s can be given by

$$n_s = \frac{Nd}{4N} = \frac{\bar{k}}{2} \tag{13}$$

Exact analysis of the arrival process at network channels is intractable because the superposition and splitting of the traffic prevalently occurs when traffic stream enters and leaves a channel. The arrival process at network channels is approximated by an MMPP(2), which is chosen such that several of its statistical characteristics identically match those of traffic generated by n_s sources. This approximation method has been commonly accepted for analysing the traffic characteristic resulting from superposition of more than one bursty sources [7], [8], [12].

To derive the characteristics of the MMPP(2) representing the traffic at network channels, we follow the method suggested by Heffes and Lucantoni [7]. In this method, the four parameters of the MMPP are chosen so that the following characteristics of the arrival traffic at a channel are matched: *i)* the mean arrival rate; *ii)* the variance-to-mean ratio of the number of arrivals in some time interval; *iii)* the long term variance-to-mean ratio of the number of arrivals; *iv)* the third moment of the number of arrivals in some time interval.

Let the subscript *c* relate a given quantity of the arrival traffic to a network channel. The above four characteristics of the superposed traffic generated by n_s sources can be written as [7], [9]

$$\lambda_c = n_s \lambda_s \tag{14}$$

$$\frac{Var[N_c(t)]}{E[N_c(t)]} = \frac{Var[N_s(t)]}{E[N_s(t)]} \tag{15}$$

$$\lim_{t\to\infty} \frac{Var[N_c(t)]}{E[N_c(t)]} = \lim_{t\to\infty} \frac{Var[N_s(t)]}{E[N_s(t)]} \tag{16}$$

$$E[(N_c(t) - E[N_c(t)])^3] = n_s E[(N_s(t) - E[N_s(t)])^3] \tag{17}$$

With the above parameters expressed by equations 14~17 as input parameters, the algorithm proposed in [7] derives the infinitesimal generator $\mathbf{Q}_c$ and the rate matrix Λ_c of the MMPP(2) that closely matches the characteristics of the traffic arriving at a given network channel. $\mathbf{Q}_c$ and Λ_c are given by

$$\mathbf{Q}_c = \begin{bmatrix} -\delta_1 & \delta_1 \\ \delta_2 & -\delta_2 \end{bmatrix}, \ \Lambda_c = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{18}$$

2.2. Calculation of the Laplace-Stieltjes Transforms of the Blocking Time ($B^*(s)$)

To determine the waiting time, w_b, to acquire a virtual channel a physical channel is treated as an MMPP/G/1 queue. Since message destinations are uniformly distributed across the network nodes, with adaptive routing the service time seen by message at network channels is the same and is equal to *t* [11], (whose Laplace-Stieltjes transform is given by equation 6). The mean of the waiting time at customer arrival instants, $\overline{w_b}$, is given as [6]

$$\overline{w_v} = \frac{1}{2(1-\rho)}\left[2\rho + \lambda_c \bar{t}^{(2)} - 2\bar{t}((1-\rho)\mathbf{g} + \bar{t}\pi\Lambda_c)(\mathbf{Q}_c + \mathbf{e}\pi)^{-1}\tilde{\lambda}\right]$$

$$\overline{w_b} = \frac{1}{\rho}\left(\overline{w_v} - \frac{1}{2}\lambda_c \bar{t}^{(2)}\right) \tag{19}$$

In the above equations, $\bar{t}$, $\bar{t}^{(2)}$ and $\bar{t}^{(3)}$ denote the mean, second and third moment of service time at a network channel, respectively. These quantities are given by differentiating $T^*(s)$ and setting $s = 0$ [9]. The traffic intensity, ρ, is given by $\rho = \bar{t}\lambda_c$. π is the steady-state vector and is written as $\pi = \frac{1}{\delta_1 + \delta_2}(\delta_2, \delta_1)$, while $\tilde{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$, $\mathbf{e} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

The algorithm to calculate matrix $\mathbf{g}$ is described in [6]. From the algorithm, we can find also that the Laplace-Stieltjes transform, $T^*(s)$, is required to computer $\mathbf{g}$.

In order to derive $T^*(s)$ described by equation 6, we need calculate the Laplace-Stieltjes transform, $B^*(s)$, of the distribution of the blocking time. However, finding the exact expression of the distribution function of the blocking time is very difficult and complex undertaking. Given that any distribution function can be approximated as closely as desired by a series-parallel stage-type of exponential distributions [9], the distribution of the blocking time can be reasonably approximated by an exponential distribution. Therefore, we can approximately express the probability density function of the blocking time, $B(x)$, and its Laplace-Stieltjes transform as

$$B(x) = \beta e^{-\beta x} \qquad (\alpha > 0) \tag{20}$$

$$B^*(s) = \frac{\beta}{s + \beta} \tag{21}$$

where β are selected to match the mean, $\bar{b}$, of the blocking time. β is found to be

$$\beta = 1/\bar{b} \tag{22}$$

From the definition of mean along with equation 5, $\bar{b}$ can be easily expressed using $\overline{w_b}$ obtained from equation 19.

2.3. Calculation of the Probability of Blocking (P_{b_i})

The probability that a message is blocked at a given node depends on its current network position. This is because the number of alternative paths that a message can take to advance towards its destination is determined by the number of the remaining hops and the ways to distribute these hops in each dimension. For a message having made *i* hops, we denote *x, y* as the numbers of hops achieved in the first and second dimensions respectively, where $(x + y = i), (0 \le x, y \le \bar{k})$. To determine the probability that a message has crossed all the channels in one dimension, two cases need to be considered.

i) When $0 \le i < \bar{k}$, the message has not crossed any dimension since it has to make $\bar{k}$ hops along each dimension. As a result, it still can choose among virtual channels of both dimensions.

ii) When $\bar{k} \le i < d$, the number of ways to distribute the hops along the two dimensions is $(d - i + 1)$. In only two cases, $(x = \bar{k}, y = i - \bar{k})$ and $(x = i - \bar{k}, y = \bar{k})$, has a message crossed all channels of one dimension, and thus all the remaining hops have to be made in another dimension.

So, the probability that a message can choose virtual channels in only one dimension after making *i* hops can be written as

$$P_{\theta_i} = \begin{cases} 0 & 0 \le i < \bar{k} \\ 2/(d - i + 1) & \bar{k} \le i < d \end{cases} \tag{23}$$

A message is blocked when all the virtual channels belonging to class *a* in the remaining dimensions to be visited and also the virtual channels belonging to class *b* in the lowest dimension to be visited are busy [2], [4], [11]. The probability, P_{b_i}, that a message is blocked after making *i* hops can be written as

$$P_{b_i} = P_{\theta_i} P_{a\&b} + (1 - P_{\theta_i}) P_a P_{a\&b} \qquad (24)$$

with P_a being the probability that all virtual channels belonging to class *a* in a physical channel are busy and $P_{a\&b}$ being the probability that all virtual channels belonging to class *a* and class *b* in a physical channel are busy. Let P_V be the probability that V virtual channels at a given physical channel are busy (P_V is determined below in Section 2.5). P_a and $P_{a\&b}$ are given in terms of P_V as (see [11] for more details on the calculation of these probability)

$$P_a = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} + \frac{P_{V-2}}{\binom{V}{V-2}} \qquad (25)$$

$$P_{a\&b} = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} \qquad (26)$$

2.4. Calculation of the Mean Waiting Time at the Source ($\overline{w}_s$)

To determine the mean waiting time, $\overline{w}_s$, that a message encounters in the source node before entering the network, the injection channel is modelled as an MMPP/G/1 queueing system. The derivation of $\overline{w}_s$ is similar to that used for the mean waiting time, $\overline{w}_b$, at a network channel. The traffic generated at the local queue is characterised by the infinitesimal generator $\mathbf{Q}_S$ and the rate matrix Λ_S (given by equation 7). Equation 19 is used to compute the mean waiting time at the source.

2.5. Calculation of the Average Degree of Virtual Channels Multiplexing ($\overline{V}$)

The probability, P_φ $(0 \le \varphi \le V)$, that φ virtual channels at a given physical channel are busy, can be determined using a Markovian model [3]. State V_φ corresponds to φ virtual channels being busy. The transition rate out of state V_φ to $V_{\varphi+1}$ is λ_c, where λ_c is the average traffic rate on a given channel (and is calculated by equation 14), while the rate out of V_φ to $V_{\varphi-1}$ is $1/\bar{t}$. The transition rate out of the last state, V_V, is reduced by λ_c to account for the arrival of messages while a channel is in this state. In the steady state, the model yields the following probabilities.

$$q_\varphi = \begin{cases} 1 & \varphi = 0 \\ q_{\varphi-1}\lambda_c \bar{t} & 0 < \varphi < V \\ q_{\varphi-1}\lambda_c \big/ (1/\bar{t} - \lambda_c) & \varphi = V \end{cases} \qquad (27)$$

$$P_\varphi = \begin{cases} 1 \Big/ \sum_{l=0}^{V} q_l & \varphi = 0 \\ P_{\varphi-1}\lambda_c \bar{t} & 0 < \varphi < V \\ P_{\varphi-1}\lambda_c \big/ (1/\bar{t} - \lambda_c) & \varphi = V \end{cases} \qquad (28)$$

In virtual channel flow control, multiple virtual channels share the bandwidth of a physical channel in a time-multiplexed manner. The average degree of multiplexing of virtual channels, that takes place at a given physical channel, is given by [3]

$$\overline{V} = \frac{\sum_{\varphi=0}^{V} \varphi^2 P_\varphi}{\sum_{\varphi=0}^{V} \varphi P_\varphi} \qquad (29)$$

3. VALIDATION OF THE MODEL

The above model has been validated by means of a discrete-event simulator. Figures 1~2 depict mean message latency results predicted by the above model plotted against those provided by the simulator versus offered traffic in the torus networks under the following cases: network size $N = 16^2$ and 24^2 nodes; number of virtual channels $V = 3$ and 5; message length $M = 32$ and 48 flits. The infinitesimal generator $\mathbf{Q}_S$ denotes the different degree of burstiness in traffic generated by source nodes. In all the figures presented below, the *x*-axis represents the traffic rate, μ_1, at which a node injects messages into the network when the IPP input traffic is at state 1. The *y*-axis gives the mean message latency. The figures reveal that the simulation results match those predicted by the analytical model in the steady state region. Its simplicity makes it a practical and cost-effective evaluation tool.

4. CONCLUSION

Before the domain of multicomputers can be fully expanded to encompass multimedia applications, it is necessary to re-examine the performance properties of their interconnection network in the context of these emerging applications in order to meet their communication requirements. This paper proposes a new analytical model for adaptive wormhole-routed torus networks with IPP input traffic. Results from simulation experiments have revealed that the model predicts message latency with a good degree of accuracy. The next step in our work is to develop a model for other well-known switching method including circuit switching and packet switching.

5. REFERENCES

[1] A. Agarwal, Limits on interconnection network performance. IEEE Trans. Parallel & Distributed Systems, vol. 2(2), pp. 398-412, 1991.

[2] Y.M. Boura, C.R. Das, T.M. Jacob, A performance model for adaptive routing in hypercubes. Proc. 1st Int. Workshop Parallel Processing, pp. 11-16, 1994.

[3] W.J. Dally, Virtual channel flow control. IEEE Trans. Parallel & Distributed Systems, vol. 3(2), pp. 194-205, 1992.

[4] J. Duato, A New theory of deadlock-free adaptive routing in wormhole routing networks. IEEE Trans. Parallel & Distributed Systems, vol. 4(12), pp. 1320-1331, 1993.

[5] J. Duato, S. Yalamanchili, L. Ni, Interconnection networks:

An engineering approach. IEEE Computer Society Press, 1997.

[6] W. Fischer, K. Meier-Hellstern, The Markov-modulated Poisson process (MMPP) cookbook. Performance Evaluation, vol. 18(2), pp. 149-171, 1993.

[7] H. Heffes, D. M. Lucantoni, A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. IEEE J. Select. Areas Com., vol. 4(6), pp. 856-868, 1986.

[8] S. H. Kang, D. K. Sung, B. D. Choi, CAC scheme based on real-time cell loss estimation for ATM multiplexers. IEEE Trans. Communications, vol. 48(2), pp. 252-258, 2000.

[9] L. Kleinrock, Queueing Systems. vol. 1, John Wiley & Sons, New York, 1975.

[10] G. Min, H. Sarbazi-Azad, M. Ould-Khaoua, Performance modelling of pipelined circuit switching in multicomputer networks. Proc. 8th Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'2000), IEEE Computer Society Press, pp. 299-306, 2000.

[11] M. Ould-Khaoua, A performance model for Duato's fully adaptive routing algorithm in k-ary n-cubes. IEEE Trans. Computers, vol. 48(12), pp. 1-8, 1999.

[12] S. S. Wang, J. A. Silvester, An approximate model for performance evaluation of real-time multimedia communication system. Performance Evaluation, vol. 22(2), pp. 239-256, 1995.

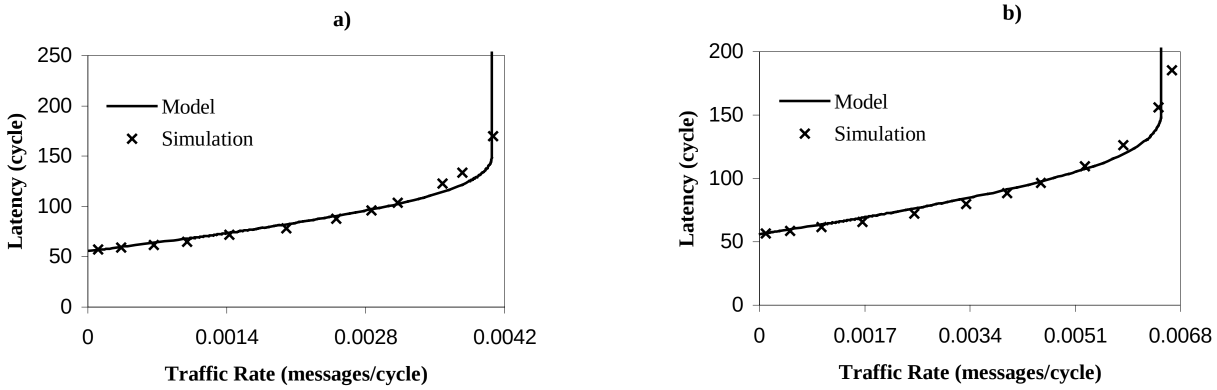

Figure 1: Latency predicted by the model and simulation in the torus network, $N = 16^2$, $V = 3$, $M = 48$ flits, a) $\sigma_1 = 0.2, \sigma_2 = 0.8$ and b) $\sigma_1 = 0.6, \sigma_2 = 0.6$.

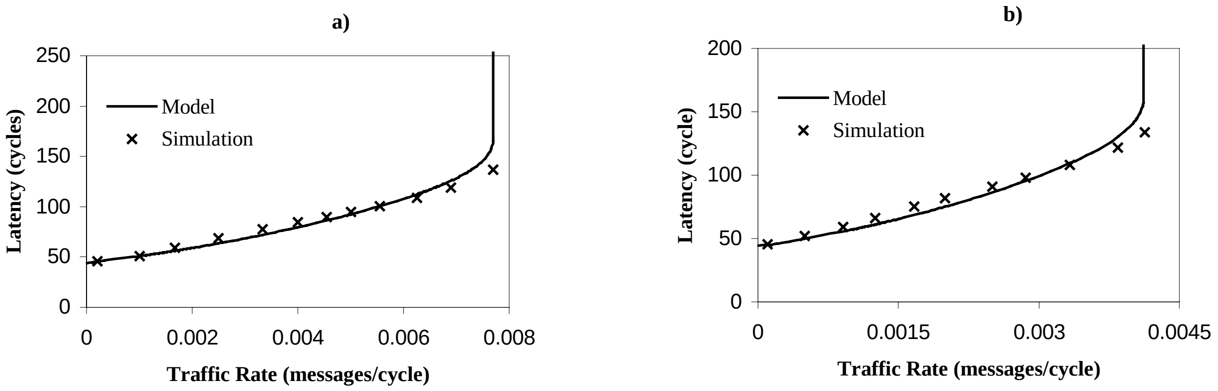

Figure 2: Latency predicted by the model and simulation in the torus network, $N = 24^2$, $V = 5$, $M = 32$ flits, a) $\sigma_1 = 0.09, \sigma_2 = 0.09$ and b) $\sigma_1 = 0.05, \sigma_2 = 0.8$.