Performance Modeling of Fault-Tolerant Circuit-Switched Communication Networks*

F. Safaei[1,3], A. Khonsari[2,1], M. Fathy[3], N. Alzeidi[4], M. Ould-Khaoua[4]

[1] IPM School of Computer Science, Tehran, Iran
[2] Dept. of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[3] Dept. of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[4] Dept. of Computing Science, University of Glasgow, UK
{safaei, ak}@ipm.ir, {f_safaei, mahfathy}@iust.ac.ir, {zeidi, mohamed}@dcs.gla.ac.uk

Abstract

Circuit Switching (CS) has been suggested as an efficient switching method for supporting simultaneous communications (such as data, voice, and images) across parallel systems due to its ability to preserve both communication performance and fault-tolerant demands in such systems. In this paper we present an efficient scheme to capture the mean message latency in 2-D torus with CS in the presence of faulty components. We have also conducted extensive simulation experiments, the results of which are used to validate the analytical model.

1. Introduction

Large-scale parallel computers, Multiprocessors System-on-Chip (MP-SoCs), multicomputers, cluster computers and peer-to-peer networks are considered today a very promising approach of achieving high computational power. In the past few years there have been tremendous improvements of parallel computers, but these have come primarily from advances in processor technology. Communication networks have not kept up, and the increasing disparity between processor and network speeds has become a major hindrance to further performance gains [1].

In addition to the impacts of communication systems on the performance, it may also account for a significant fraction of the cost and power dissipation of such systems. Ideally, the communication systems should deliver messages with low latency and support high throughput and should also be cost-effective, reliable and *fault-tolerant*. A communication network can be defined by its *topology*, *routing*, and *switching*.

The network topology defines the connectivity of the communication network. The torus (also known as *k*-ary 2-cube) has become a widely accepted communication network due to its desirable and powerful topological properties.

Routing determines the path a message will take through the network. Routing algorithms for large-scale parallel computers are generally classified as being either *deterministic* or *adaptive*. Deterministic routing is simple and easily implemented, with minimal overhead. Adaptive routing, on the other hand, improves both the performance and fault-tolerance of a communication network and, more importantly, allows for further flexibility at the cost of additional complexity in the algorithm and its implementation [2].

The switching technique determines how messages are propagated from the source to the destination, including the hardware protocols for transmitting data across a physical channel and for buffering data at a router [1]. Recently there has been a renewed interest for Circuit Switching (CS) in the communication systems and Internet [3, 4]. The main reason is that it is now possible to build extremely high capacity all-optical switches using this technology, whereas it is not possible to design them using packet switching. This is because routers (i.e. packet switches) need to buffer packets for contention resolution, and we still do not know how to store photons while providing random access as we do with electronic RAMs. Circuit switches are characterized by simple data paths requiring no per-packet processing, and no packet buffers. Moreover, there are a few other advantages to CS. For relatively long messages, the message latency is almost independent of the distance between the source and destination. In addition, there are no storage requirements at the routers, since no data is sent until the circuit has been established. The other advantage of CS is its ability to provide messages with an agreed-upon *Quality of Services* (QoS), e.g., guaranteed latency, once a connection has been established [3-5].

* This research was in part supported by a grant from I.P.M. (No. CS1384-3-01).

These features make circuit-switched networks suitable for supporting simultaneous (data, voice and image) communications across parallel computers, distributed computers, and telecommunication systems.

In recent years, many studies have addressed several issues in the field of fault-tolerance and reliability analysis of large-scale parallel and distributed systems. These researches span a wide range of systems such as massively parallel processors, cluster-based systems, mobile systems, sensor networks, and more recently Mp-SoCs. Fault-tolerant designs of these systems aim at providing continuous operations in the presence of faults by allowing the graceful degradation of system. Almost all of the proposed methods and algorithms for functionality of the recent systems have resorted to simulation to study the performance of such systems, under different failure conditions. The limitations of simulation-based solutions are that it is highly time-consuming and expensive. An alternative to simulation approach is an analytical model, which is the focus of this research. The significant advantage of analytical models over simulation is that they can be used to obtain performance results for large systems which may not feasible to study using simulation due to the excessive computation demands. An accurate analytical model can provide quick performance estimates and will be a valuable design tool.

Several models analyzing CS have been proposed in the literature [4, 6-8]. However, the performance properties of CS have not been thoroughly investigated in the presence of faulty components. Due to recent popularity of CS in providing QoS in the networks, the study presented in this paper evaluates the performance, both analytically and experimentally, of a CS on the torus networks with faults. The goals of our study are to evaluate the performance potential of CS to understand the effects of failures on switching and routing performance.

The rest of the paper is organized as follows. Necessary definitions are introduced in Section 2. The proposed analytical model is derived and discussed in Section 3. In addition to the theoretical analysis, extensive simulation experiments were performed. The results of these experiments for validation of the mathematical model are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Torus network topology and its node structure

The topology we have chosen to investigate is the 2-dimensional torus (*k*-ary 2-cube). This is a topology which has become increasingly popular among researches and has been implemented in a number of existing machines [1, 5]. A *k*-ary 2-cube is a direct network with $N=k^2$ nodes; k is called the radix. Links (channels) in the torus can be either uni- or bi-directional. In this paper, we will focus on 2-D torus with bi-directional links as they have been more popular in parallel systems. Each node can be identified by a 2-digit radix k address (a_1, a_2). Nodes, with address (a_1, a_2), (b_1, b_2) are connected if and only if $a_1 = (a_2 + 1) \bmod k$ or $b_1 = (b_2 + 1) \bmod k$. In order to allow processors to concentrate on computational tasks and permit the overlapping of communication with computation, a *router*, is used for handling message communication among processors, and is usually associated with each processor. Consequently, each node consists of a Processing Element (PE) and router. A node is connected to its neighboring nodes via the input and output channels. The *injection/ejection* channel is used by the processor to inject/eject messages to/from the network.

3. Analysis

This section describes the assumptions used in the analysis, and then presents the analytical model.

3.1 Assumptions

The analytical model is based on the following assumptions that have been widely used in literature [4, 6-15].

1. Each processor generates messages independently, which follows a Poisson process with a mean rate of λ_g.
2. The arrival process at a given communication network is approximated by an independent Poisson process. Therefore, the rate of process arrival at a communication network can be calculated using *Jackson's queuing networks* formula [10].
3. The destination of each message would be any node in the network with uniform distribution.
4. Message length is M flits, where M is a random variable whose first, $\bar{M}$, and second moment, $\overline{M^2}$, are known. Each flit requires one-cycle transmission time across a physical channel.
5. The local queue at the injection channel in the source node has infinite capacity. Moreover, messages are transferred through the ejection channel to the local node as soon as they arrive at their destinations.
6. Each node failed with probability θ. The probabilities of node failure in the network are equiprobable and independent of each other. Moreover, Faults are static [1, 5] and distributed

uniformly throughout the network such that do not disconnect the network.

7. Nodes (processors) are more complex than links and thus have higher failure rates [1, 5]. Therefore, we focus only on node failures.
8. V virtual channels $(V \geq 1)$ are used per physical channel. When there is more than one virtual channel available that bring a message closer to its destination, one is chosen at random.
9. If the header is blocked upon reaching an intermediate node, the partial path is torn down by propagating a "release" signal backward to the source to release all the acquired channels. Then, the header backtracks to the source and makes a new attempt to set up a connection [4, 5, 8].

3.2 Outline of the analytical model

In this section, we present the mathematical model that approximate the behavior of 2-D torus communication system using CS in the presence of faulty components.

The mean message latency is composed of the mean network latency, $\bar{T}_r$, which is the time to cross the network and the mean waiting time seen by the message in the source node, $\bar{W}_s$, before entering the network. $\bar{T}_r$ includes the delay experienced by a message in the previous re-transmission attempts as well. However, to capture the effects of virtual channels multiplexing, the mean message latency has to be scaled by a factor, say $\bar{V}$, representing the average degree of virtual channels multiplexing, that takes place at a given physical channels. Therefore, the mean message latency can be approximated as [11]

$$Latency = (\bar{T}_r + \bar{W}_s)\bar{V} \quad (1)$$

In what follows, we will describe the calculation of $\bar{T}_r$, $\bar{W}_s$, and $\bar{V}$.

3.2.1 Calculation of the mean network latency

Under the uniform traffic pattern, the average number of channels that a message visits along a given dimension and across the network, $\bar{k}$, $\bar{d}$ respectively, are given by Agarwal [12]

$$\bar{k} \approx k/4, \ \bar{d} = 2\bar{k} \quad (2)$$

Consider the header that has successfully established a connection because it has not encountered a connection failure at any intermediate channels. The service time seen by a message at a given channel in the case of successful connection consists of the time required by the header to establish a connection and the time to transmit the message. If T_S is the random variable denoting the service time in the case of successful connection, we can write

$$T_S = M + 2\bar{d} \quad (3)$$

Where M is a random variable denoting the message length. Note that in Equation (3) the term $2\bar{d}$ accounts for $2\bar{d}$ cycles that are required to send the acknowledgement flit back to the source node. The Laplace-Stieltjes [10] transform of T_s can be written as

$$L^*_{T_S}(S) = L^*_M(S) \cdot e^{-2S\bar{d}} \quad (4)$$

Consider the header that encounters faulty/blocking situation, and as a result experiences a connection failure at the j-th $(1 \leq j \leq \bar{d})$ hop channel. A negative acknowledgement is then propagated backward to the preceding node, requiring a single cycle at each hop. The service time seen by a message ant its Laplace-Stieltjes transform in the case of a connection failure that occurred at the j-th hop channel are simply

$$T_{F_j} = 2(j-1) \quad (5)$$

$$L^*_{T_{F_j}}(S) = e^{-2S(j-1)} \quad (6)$$

Let ξ_j be the number of physical channels that the header can select at its j-th hop to advance towards its destination. The header is blocked at this channel when encounters a faulty component or finds that all of the V virtual channels at each of the ξ_j physical channels are busy. Since adaptive routing distributes traffic equally among network channels the probability, P_v, that v virtual channels at a given physical channel are busy, is the same at all network channels regardless of their positions. Given that the header reaches the j-th hop if it has not suffered a connection failure at the (j-1) channels, the probability, P_{F_j}, that it suffers a connection failure at the j-th hop channel can be expressed as

$$P_{F_j} = (P_V + \theta)^{\xi_j} \prod_{i=1}^{j-1} \left(1 - (P_V + \theta)^{\xi_i}\right) \quad ; P_V + \theta < 1 \quad (7)$$

A connection failure can be caused by blocking at any of the $\bar{d}$ channels along the network path. Removing the conditioning on the channel where the blocking has occurred, we can write the service time at a given channel in the case of connection failure as

$$L^*_{T_F}(S) = \sum_{j=1}^{\bar{d}} P_{F_j} L^*_{T_{F_j}}(S) \quad (8)$$

Let P_S be the probability of a successful connection, and P_F be the probability of a connection failure. Since

the header crosses, on average, $\bar{d}$ channels to reach its destination, P_S and P_F are given by

$$P_S = \prod_{i=1}^{\bar{d}-1}\left(1-(P_V+\theta)^{\xi_i}\right) \quad (9)$$

$$P_F = 1-\prod_{i=1}^{\bar{d}-1}\left(1-(P_V+\theta)^{\xi_i}\right) \quad (10)$$

Therefore, the weighted service time seen by a message at a given channel, taking into account the cases of connection success and failure, is given by

$$\begin{aligned} &L_T^*(S) = P_S L_{T_S}^*(S) + \sum_{j=1}^{\bar{d}} P_{F_j} L_{T_{F_j}}^*(S) = \\ &\prod_{j=1}^{\bar{d}-1}\left(1-(P_V+\theta)^{\xi_j}\right) L_M^*(S) e^{-2s\bar{d}} + \\ &\sum_{j=1}^{\bar{d}}\left((P_V+\theta)^{\xi_j}\prod_{i=1}^{j-1}\left(1-(P_V+\theta)^{\xi_i}\right)e^{-2s(i-1)}\right) \end{aligned} \quad (11)$$

Consider a message that required n retrials to successfully set-up a connection because it previously experienced (n-1) connection failures. Let T_n be a random variable denoting the service time seen by a message at a given channel after n retrials. Moreover, $L_{T_n}^*(S)$ be the corresponding Laplace-Stieltjes transform. We can write T_n as

$$T_n = \sum_{n=1}^{\infty} t\cdot P_S + (n-1)t\cdot P_F^{n-1} \quad (12)$$

$$\begin{aligned} &L_{T_n}^*(S) = \sum_{n=1}^{\infty} P_S P_F^{n-1}\left(L_T^*(S)\right)^n = \\ &P_S L_T^*(S)\sum_{n=0}^{\infty} P_F^{n-1}\left(L_T^*(S)\right)^{n-1} = P_S L_T^*(S)\Big/\left(1-P_S L_T^*(S)\right) \end{aligned} \quad (13)$$

After the header has successfully established a path between source and destination nodes, the first data flit takes $\bar{d}$ cycles to reach the destination. Therefore, the mean network latency seen by a message to cross from source to destination is given by

$$\bar{T}_r = \bar{T}_n + \bar{d} \quad (14)$$

Where $\bar{T}_n$ is the mean service time seen by a message at a given channel, and is obtained as

$$\bar{T}_n = E[T_n] = \frac{\partial}{\partial S} L_{T_n}^*(S)\Big|_{S=0} = (P_S T_S + \sum_{j=1}^{\bar{d}} P_{F_j} T_{F_j})\Big/P_S \quad (15)$$

To compute the number of channels, ξ_j, that a message can select when crossing the j-th hop channel, $(1 \le j \le \bar{d})$, we use the method described in [15]. We recollect briefly here the main equations for the calculation of ξ_j. The number of channels that the message can select when crossing the j-th hop channel is given by

$$\xi_j = \sum_{t=0}^{2}(2-t)\,\psi_j^t \quad (16)$$

Where ψ_j^t is the probability that the header has entirely crossed t dimensions along on its j-hop path. This probability is a function of the number of dimensions that the message has still to cross. The probability that there remains only one dimension to cross a message j-hops away from its destination, P_{φ_j}, can be written as

$$P_{\varphi_j} = \begin{cases} 2/(\bar{d}-j+1) & \bar{k} \le j < \bar{d}-1 \\ 0 & 0 \le j < \bar{k} \end{cases} \quad (17)$$

Consequently, the probability that the header has entirely crossed t dimensions along on its j-hop path is given by

$$\psi_j^t = \begin{cases} 1-P_{\varphi_j} & t=0 \\ P_{\varphi_j} & t=1 \end{cases} \quad (18)$$

3.2.2 Calculation of the mean waiting time in the source node

In this section, we compute the mean waiting time at the source node ($\bar{W}_s$). Since a message in the source node can enter the network through any of the V virtual channels, the mean arrival rate to the queue is λ_g/V. Since a message does not leave the first hop channel, and therefore the local queue, until a connection has been established through the network, the service time seen by a message is exactly $L_{T_n}^*(S)$ (given by Equation (13)). Applying the Pollaczek-Khinchine (P-K) mean value formula [10] yields the mean waiting time experienced by a message at the source node as [10]

$$\bar{W}_s = (\lambda_g/V)\,E[T_n^2]\Big/2\left(1-(\lambda_g/V)E[T_n]\right) \quad (19)$$

Where $E[T_n]$ and $E[T_n^2]$ are the first and second moment of T_n. While $E[T_n]$ has already been computed in Equation (15), parameter $E[T_n^2]$ can be expressed as follows

$$E[T_n^2] = \frac{\partial^2}{\partial S^2} L_{T_n}^*(S)\Big|_{S=0} = -\left(P_S E[T_n^2] + 2P_F E^2[T_n]\right)\Big/P_S^2 \quad (20)$$

The probability, P_v ($0 \le v \le V$), that v virtual channels at a given physical channel are busy can be determined using a Markovian model (details of the model can be found in [14]). In the steady-state, the model yields the probability P_v as [14]

$$q_0 = 1 \quad (21)$$

$$q_v = q_{v-1}\lambda_c \bar{T}_r \qquad (1 \le v \le V-1) \quad (22)$$

$$q_V = q_{V-1}\lambda_c / (1/\overline{T}_r - \lambda_c) \quad (23)$$

$$P_0 = \left(\sum_{v=0}^{V} q_v\right)^{-1} \quad (24)$$

$$P_v = P_{v-1}\lambda_c \overline{T}_r \qquad (1 \le v \le V-1) \quad (25)$$

$$P_V = P_{V-1}\lambda_c / (1/\overline{T}_r - \lambda_c) \quad (26)$$

When multiple virtual channels are used per physical channel they share the bandwidth in a time multiplexed manner. The average degree of virtual channel multiplexing, that takes place at a given physical channel, can be estimated by [14]

$$\overline{V} = \sum_{v=1}^{V} v^2 P_v \Big/ \sum_{v=1}^{V} v P_v \quad (27)$$

The mean arrival rate, λ_c, on a given channel is determined as follows. A PE generates, on average, λ_g messages in a cycle, which are evenly distributed among the 4 output channels. Each message may have to be retransmitted a number of times before it successfully reaches its destination. By summing up the rate of messages generated by the PE and those due to transmission failures, we can write the *effective* traffic rate, λ_e, offered to an output channel as

$$\lambda_e = \sum_{r=0}^{\infty} P_F^r \lambda_g / 4 = \lambda_g / 4P_S \quad (28)$$

In addition to messages generated by the local PE, a given channel may receive messages from PEs that are within 1, 2, to $(\overline{d}-1)$ hops away. These message headers cross the channel if they have not experienced a transmission failure before reaching the channel. Therefore, the traffic rate, λ_c, on a channel is given by

$$\lambda_c = \sum_{j=0}^{\overline{d}-1} \prod_{i=1}^{j} \left(1 - (P_V + \theta)^{\xi_j}\right) \lambda_e \quad ; P_V + \theta < 1 \quad (29)$$

4. Experimental results

A discrete-event simulator was developed in order to verify the mathematical analysis and to further understand and evaluate the performance of CS with faults in the 2-D torus when fully adaptive routing and virtual channels flow control are used. All measurements were taken only after the system had reached steady state. Moreover, in order to minimize sampling errors, the latencies were generated by averaging the results of multiple runs. Extensive validation experiments have been performed for several combinations of network sizes, message lengths, failure rates, and virtual channels. However, for the sake of specific illustration, latency results are presented for the following cases only:

- Network size $N = 8^2$ and 16^2 nodes.
- Number of virtual channels $V = 1$, 10 per physical channel.
- Mean message length $M = 32$ and 64 flits.
- Failure rates $\theta = 0.0$, 0.1, 0.2.

4.1 Model validation

The average message latencies obtained from the simulation and analytical results plotted in Figure 1 for a varying number of virtual channels and different failure rates. The *x*-axis in this figure represents the traffic rate injected by a given in a cycle (λ_g) while the *y*-axis shows the mean message latency (in cycles). The figure indicates that the simulation results and the values predicted by the analytical model are in good agreement (lower than 5%) under steady state regions, i.e. under light and moderate traffic and near the saturation point. However, due to the approximations made when constructing the model, some discrepancies (of at most 15% error) are apparent around the saturation point. Since the independence assumptions are essential in ensuring a tractable model, and given that most evaluation studies concentrate on network performance in the steady state regions, it can be concluded that the present model constitutes a cost-effective evaluation tool for assessing the performance behavior of circuit-switched torus networks in the presence of faulty components.

5. Conclusions

Analytical models of fully adaptive routing have recently been proposed for CS in torus networks. However, there has not been any analytical model of CS in the presence of faulty components. This paper has presented a novel analytical model for the performance evaluation of CS in the torus in the presence of failures when fully adaptive routing and virtual channels flow control are used. The proposed model has been examined with a number of widely used cases in order to evaluate the performance of the CS under various working conditions. Simulation experiments have shown that the results predicted by the analytical model are in good agreement with those obtained through simulation experiments. Our next objective is to extend the above modeling approach to consider other well-known fault-tolerant routing algorithms.

References

[1] W.J. Dally and B. Towles, *Principles and practices of interconnection networks*, Morgan Kaufman Publishers, 2004.

[2] P. Mohapatra, Wormhole Routing Techniques for Directly Connected Multicomputer Systems, ACM Computing Surveys, Vol. 30, No. 3, September 1998.

[3] P. Molinero-Fernandez, N. McKeown, The Performance of Circuit Switching in the Internet,

Journal of Optical Networking, Vol. 2, pp. 82-96, 2003.

[4] G. Min, M. Ould-Khaoua, H. Sdazi-Azad, Communication Delay in Circuit-Switched Interconnection Networks, IPCCC 2001, pp. 51-56, 2001.

[5] J. Duato, S. Yalamanchili, L.M. Ni, *Interconnection networks: An engineering approach*, Morgan Kaufmann Publishers, 2003.

[6] L. Chlamtac, A. Ganz, M.G. Kienzle, A performance model of a connection-oriented hypercube interconnection system, Performance Evaluation, Vol. 25, No. 2, pp. 151-167, 1996.

[7] M. Colajarmi, B. Ciciani, S. Tucci, Performance analysis of circuit-switching interconnection networks with deterministic and adaptive routing, *Performance Evaluation*, Vol. 34, No. 1, pp. 1-26, 1998.

[8] V. Sharma, EA. Varvarigos, Circuit switching with input queuing: an analysis for the *d*-dimensional wraparound mesh and the hypercube, *IEEE Trans. Parallel & Distributed systems*, Vol. 8, No. 4, pp.349-366, 1997.

[9] W.J. Dally, Performance analysis of k-ary n-cubes interconnection networks, *IEEE Trans. Computers,* Vol. 39, No.6, pp. 775-785, 1990.

[10] L. Kleinrock, *Queueing Systems*, Vol. 1, John Wiley, New York, 1975.

[11] M. Ould-Khaoua, A Performance model for Duato's adaptive routing algorithm in *k*-ary *n*-cubes, *IEEE Trans. Computers*, Vol. 48, No 12, pp. 1-8, 1999.

[12] A. Agarwal, Limits on interconnection network performance, *IEEE Trans. Parallel & Distributed Systems*, Vol. 2, No. 4, pp. 398-412, 1991.

[13] J. Draper, J. Ghosh, A comprehensive analytical model for wormhole routing in multicomputers systems, Journal of Parallel and Distributed Computing (JPDC), Vol. 32, pp. 202–214, 1994.

[14] W.J. Dally, Virtual channel flow control, *IEEE Trans. Parallel & Distributed Systems,* Vol. 3, No. 2, pp. 194-205, 1992.

[15] G. Min, M. Ould-Khaoua, A Comparative Study of Switching Methods in Multicomputer Networks, Journal of Supercomputing Vol. 21, pp. 227-238, 2002.

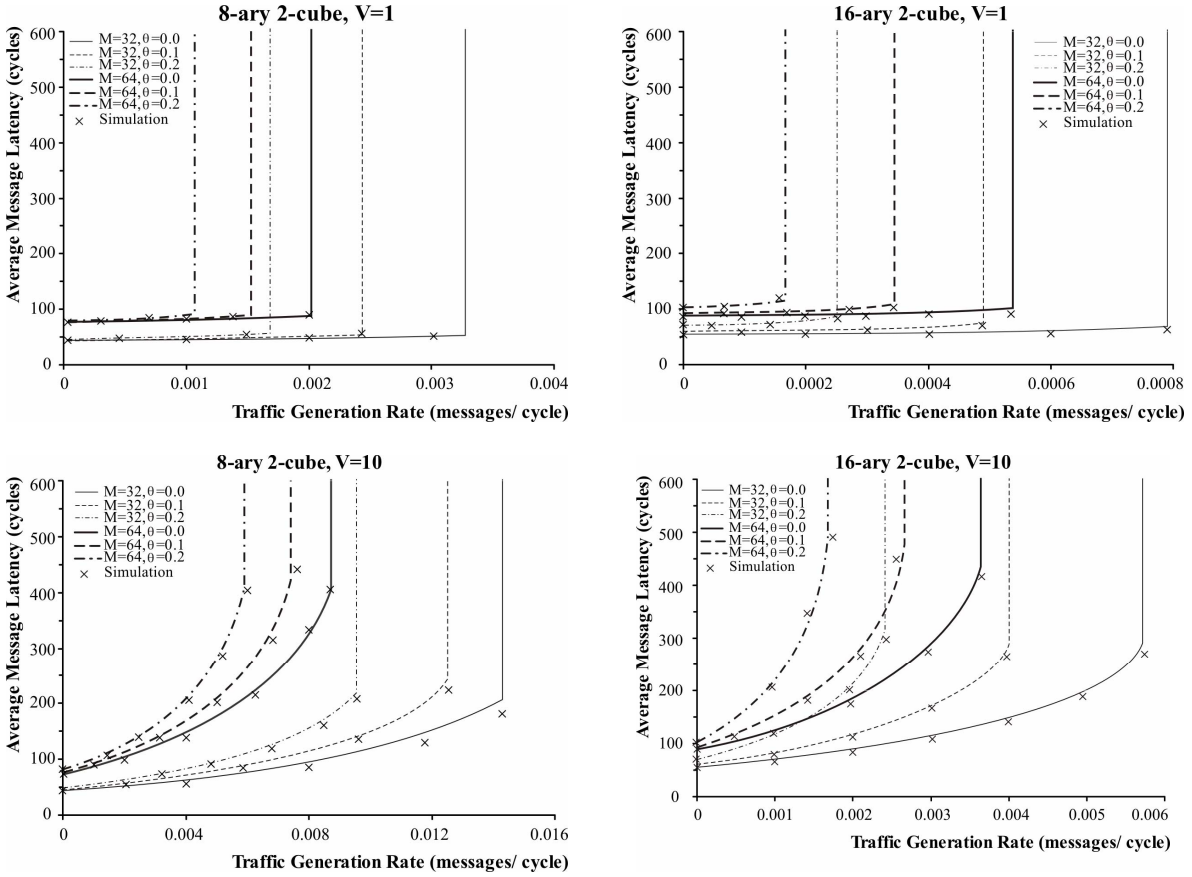

Figure 1: Average message latency calculated by the model vs. simulation in the 8×8 and 16×16 torus networks for different number of virtual channels V=1 and 10, message lengths M=32 and 64 flits and different failure rates θ =0.0, 0.1, 0.2.