Performance Evaluation of Hypercubes in the Presence of Multiple Time-Scale Correlated Traffic

Geyong Min[1] and Mohamed Ould-Khaoua[2]

[1] Department of Computing, School of Informatics, University of Bradford,
Bradford, BD7 1DP, U.K.
g.min@brad.ac.uk

[2] Department of Computing Science, University of Glasgow, Glasgow, G12 8RZ, U.K.
mohamed@dcs.gla.ac.uk

Abstract. The efficiency of a large-scale parallel computer is critically dependent on the performance of its interconnection network. Analytical modelling plays an important role towards obtaining a clear understanding of network performance under various design spaces. This paper proposes an analytical performance model for circuit-switched hypercubes in the presence of multiple time-scale correlated traffic which can appear in many parallel computation environments and has strong impact on network performance. The tractability and reasonable accuracy of the analytical model demonstrated by simulation experiments make it a practical and cost-effective evaluation tool to investigate network performance under different system configurations.

1 Introduction

Multicomputers have been widely accepted as the solution for solving grand challenge problems in high performance computing. Interconnection network [1] is a critical architectural component in multicomputer systems as any interaction between the processors ultimately depends on its effectiveness. The hypercube has been one of the popular network topologies in multicomputers owing to its desirable properties, such as regular structure, symmetry, low diameter and high connectivity to deal with fault-tolerance [2]. An *n*-dimensional hypercube has $N = 2^n$ nodes with 2 nodes in each dimension. Each node consists of a processing element and a router.

The switching strategy determines how data in a message traverses its route from source to destination. The circuit switching has been widely employed in computer and telecommunication systems [1]~[7]. Such a switching strategy is divided into two phases: (1) circuit establishment phase; (2) transmission phase. A dedicated path is set up prior to the transmission of data. A noticeable advantage of circuit switching is due to the fact that it does not require packetizing. Moreover, the low buffer requirement enables the construction of small, compact, and fast routers [1].

[1] The research was supported in part by the Nuffield Foundation through Grant NAL/00682/G, and the University of Bradford through Research Investment Fund.

Traffic loads generated by real-world applications have very strong effects on the performance of interconnection networks. Many recent studies [8]~[10] have demonstrated that realistic traffic can reveal burstiness and correlations among inter-arrival intervals over a number of time scales. At every time scale, traffic bursts consist of bursty subperiods separated by less bursty subperiods. This fractal-like behaviour of network traffic can be much better modelled using statistically long-range dependent processes, which reveal totally different theoretical properties from the conventional Poisson process [9]. A stochastic process X with autocorrelation function $r(k)$ is long-range dependent if its autocorrelation decays hyperbolically fast, i.e. $r(k) \sim |k|^{-\beta}$, as $|k| \to \infty$ with $0 < \beta < 1$ [11]. The Hurst parameter, $H = 1 - \beta / 2$ where $0.5 < H < 1$, is commonly used to measure the degree of long-range dependence.

Sahuquillo *et al.* [10] have traced some typical parallel applications and revealed that workloads generated by many scientific and engineering computations exhibit the fractal-like nature. In an effort towards providing cost-effective tools that help investigating the network performance with various design alternatives and under different traffic conditions, this paper proposes an analytical model for hypercube networks with circuit switching in the presence of multiple time-scale bursty and correlated traffic. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments of the actual system.

The rest of this paper is organized as follows. Section 2 presents the derivation of the analytical model. Section 3 validates the model through simulation experiments. Finally, Section 4 concludes this study.

2 Derivation of the Performance Model

The analytical model is based on the following assumptions [2], [4]~[7], [12], [13].

1) Traffic generated by each node follows an independent stochastic process with a mean arrival rate λ and autocorrelation at lag 1 $r(1) = \rho\lambda$. Traffic burstiness and correlations appear over t time scales. The autocorrelation decays hyperbolically with Hurst parameter H as the time scale increases.

2) Message destination nodes are uniformly distributed across the network nodes. Message length is M flits.

3) The local queue in the source node has infinite capacity. Each physical channel is divided into V virtual channels [1].

4) Messages are routed adaptively through the network [1, 6] using one of the available shortest paths.

Under the uniform traffic pattern, the average message distance in an *n*-dimensional hypercube is given by $D \cong n/2$ [2]. As the network latency consists of the time to establish a path and the time to transmit a message, it can be calculated as *T*=*E*+*D*+*M* where E represents the path set-up time. Since the Laplace-Stieltjes transform (LST) of the sum of independent random variables is equal to the product of their transforms [14], the LST of T can be written as

$$T^*(s) = E^*(s)e^{-s(D+M)} \tag{1}$$

where $E^*(s)$ denotes the LST of the time to set up a path.

Following the approach proposed in Ref. [15], traffic burstiness and correlations over multiple time scales can be modelled by the superposition of L two-state Markov-Modulated Poisson Processes (MMPP) [16], typically $L = 4$. We use the MMPP$^{(i)}$ with superscript i to denote the *i*-th MMPP ($1 \le i \le L$). A two-state MMPP$^{(i)}$ can be parameterised by the infinitesimal generator, $\mathbf{A}_i$, and rate matrix $\mathbf{B}_i$ as [18]

$$\mathbf{A}_i = \begin{bmatrix} -\delta_{1i} & \delta_{1i} \\ \delta_{2i} & -\delta_{2i} \end{bmatrix} \text{ and } \mathbf{B}_i = \begin{bmatrix} \lambda_{1i} & 0 \\ 0 & \lambda_{2i} \end{bmatrix} \tag{2}$$

The element δ_{1i} is the transition rate from state 1 to 2 of the MMPP$^{(i)}$ and δ_{2i} is the rate out of state 2 to 1. λ_{1i} and λ_{2i} are the traffic rate when the MMPP$^{(i)}$ is in state 1 and 2, respectively. The fitting algorithm described in Ref. [15] derives the parameters δ_{1i}, δ_{2i}, λ_{1i}, λ_{2i} for each MMPP$^{(i)}$ ($1 \le i \le L$) for matching the mean and autocorrelation function over different time scales. The superposition of the MMPP$^{(i)}$s ($1 \le i \le L$) gives rise to a new MMPP with 2^L states and its parameter matrices, $\mathbf{A}_s$ and $\mathbf{B}_s$, can be computed as (the symbol "$\oplus$" denotes the Kronecker sum [18])

$$\mathbf{A}_s = \mathbf{A}_1 \oplus \mathbf{A}_2 \oplus \cdots \oplus \mathbf{A}_L \text{ and } \mathbf{B}_s = \mathbf{B}_1 \oplus \mathbf{B}_2 \oplus \cdots \oplus \mathbf{B}_L \tag{3}$$

A message enters the network through one of the V injection virtual channels with even probability 1/*V*. Given that the process resulting from the splitting of an MMPP has the same infinitesimal generator as the original MMPP [17], the infinitesimal generator $\mathbf{A}_v$ and rate matrix $\mathbf{B}_v$, of the resulting MMPP that models the traffic on an injection virtual channel in the source node are given by

$$\mathbf{A}_v = \mathbf{A}_s \text{ and } \mathbf{B}_v = \mathbf{B}_s / V \tag{4}$$

To determine the mean waiting time that a message experiences before entering the network, the injection virtual channel in the source node is modelled as an MMPP/G/1 queueing system. The mean waiting time, Ws, can be expressed as [18]

$$Ws = \frac{1}{2\mu}\left(\frac{2\mu + \lambda_v \overline{T}^{(2)} - 2\overline{T}((1-\mu)\mathbf{g} + \overline{T}\boldsymbol{\pi}\mathbf{B}_v)(\mathbf{A}_v + \mathbf{e}\boldsymbol{\pi})^{-1}\tilde{\boldsymbol{\lambda}}}{1-\mu} - \lambda_v \overline{T}^{(2)} \right) \tag{5}$$

In the above equations, $\overline{T}$ and $\overline{T}^{(2)}$ denote the first two moments of the message service time and can be computed by differentiating $T^*(s)$ and setting $s = 0$ [14]. $\mathbf{e}$ is a unit column vector of length 2^L. The traffic intensity $\mu = \overline{T}\lambda_v$, where λ_v is the mean traffic rate and given by $\lambda_v = \boldsymbol{\pi}\tilde{\boldsymbol{\lambda}}$. $\tilde{\boldsymbol{\lambda}}$ is a column vector containing the elements on the main diagonal of $\mathbf{B}_v$, and $\boldsymbol{\pi}$ is the steady-state vector of the MMPP.

When a message header is blocked at an intermediate node, it experiences connection failure and will make a new attempt to establish a path from the source node. Let Pb_i denote the probability that the header suffers blocking after making i hops. The probability of a successful connection, Ps, and a connection failure, Pf, during a single connection attempt can be written as

$$Ps = \prod_{i=0}^{D-1}(1 - Pb_i) \quad \text{and} \quad Pf = 1 - Ps = 1 - \prod_{i=0}^{D-1}(1 - Pb_i) \tag{6}$$

A message may need a number of, say, r $(r = 1, 2, ..., \infty)$, connection attempts in order to successfully establish a path. The traffic due to the *r*-th attempt of the MMPP$^{(i)}$ $(1 \le i \le L)$ can be modelled by a new two-state MMPP$^{(ir)}$ which is the resulted process from the splitting, with the probability $Pf^{\,r-1}$, of the original MMPP$^{(i)}$. The infinitesimal generator $\mathbf{A}_{ir}$ and rate matrix $\mathbf{B}_{ir}$, of the MMPP$^{(ir)}$ is given by [17]

$$\mathbf{A}_{ir} = \mathbf{A}_i \quad \text{and} \quad \mathbf{B}_{ir} = Pf^{\,r-1}\mathbf{B}_i \tag{7}$$

Superposing the traffic caused by all r, $(r = 1, 2, ..., \infty)$, connection attempts of those generated by a source node yields the *effective* traffic entering the network. Therefore, the effective traffic can be modeled by the superposition of all MMPP$^{(ir)}$s with $(1 \le i \le L)$ and $(r = 1, 2, ..., \infty)$. As the superposition of MMPPs gives rise again to an MMPP [18], the effective traffic from a given source node can be characterised by a new multi-state MMPP. To calculate the parameter matrices of this new MMPP, we first use a two-state MMPP$^{(1e)}$ to match the superposition of all MMPP$^{(1r)}$s with $(r = 1, 2, ..., \infty)$ because these MMPPs model traffic burstiness and correlations over the same time scale. Using the parameter matrices of the MMPP$^{(1r)}$s with $(r = 1, 2, ..., \infty)$ as input parameters, the method presented in Ref. [6] for superposing infinite correlated traffic streams can be used to derive the infinitesimal generator $\mathbf{A}_{1e}$ and rate matrix $\mathbf{B}_{1e}$ of the MMPP$^{(1e)}$. Similarly, we separately match the superposition of the MMPP$^{(ir)}$s with $(r = 1, 2, ..., \infty)$ to a two-state MMPP$^{(ie)}$ with the resulting parameter matrices $\mathbf{A}_{ie}$ and $\mathbf{B}_{ie}$. We then calculate the Kronecker sum of the parameter matrices of MMPP$^{(1e)}$, MMPP$^{(2e)}$, …, MMPP$^{(Le)}$ to parameterise the multi-state MMPP that characterises the effective traffic entering the network from a given source node. So, the infinitesimal generator $\mathbf{A}_e$ and rate matrix $\mathbf{B}_e$ of the multi-state MMPP are given by

$$\mathbf{A}_e = \mathbf{A}_{1e} \oplus \mathbf{A}_{2e} \oplus \cdots \oplus \mathbf{A}_{Le} \quad \text{and} \quad \mathbf{B}_e = \mathbf{B}_{1e} \oplus \mathbf{B}_{2e} \oplus \cdots \oplus \mathbf{B}_{Le} \tag{8}$$

A message may encounter blocking at any of the D intermediate nodes along its path. The probability, Pf_i, that a header experiences a connection failure at a node that is i hops away from the source can be expressed as

$$Pf_i = Pb_i \prod_{j=0}^{i-1}(1 - Pb_j) \tag{9}$$

Taking into account the cases of a connection success and connection failures occurring at D possible nodes gives the average number of channels, c, traversed by a message during a single connection attempt

$$c = D \cdot Ps + \sum_{i=0}^{D-1} i \cdot Pf_i = \prod_{i=0}^{D-1} D(1 - Pb_i) + \sum_{i=0}^{D-1}\left(iPb_i \prod_{j=0}^{i-1}(1 - Pb_j) \right) \tag{10}$$

Under the uniform traffic pattern, using adaptive routing results in a balanced traffic load on all network channels. Examining Eq. (10) reveals that the average number of channels, c, traversed by a message during a single connection attempt is always less than n in an n-dimensional hypercube. This implies that the arrival traffic at a given network channel is a fraction of the effective traffic entering into the network from a source node. This fraction, f, can be estimated by

$$f = \frac{Nc}{Nn} = \frac{c}{n} \qquad (11)$$

Given that the MMPP is closed under the superposition and splitting operations, we use an MMPP$^{(c)}$ to model the characteristics of the traffic on a network channel. The infinitesimal generator $\mathbf{A}_c$ and rate matrix $\mathbf{B}_c$ of the MMPP$^{(c)}$ are given by [17]

$$\mathbf{A}_c = \mathbf{A}_e \quad \text{and} \quad \mathbf{B}_c = f\mathbf{B}_e \qquad (12)$$

After determining the characteristics of traffic on network channels, the joint probability $P_{(i,j)}$ that i, $(0 \le i \le V)$, virtual channels are busy and the MMPP modelling the traffic on network channels is at state j, $(1 \le j \le 2^L)$, can be calculated using a bivariate Markov chain [12]. The detailed derivation of, $P_{(i,j)}$, and calculation of the average degree of virtual channel multiplexing, $\overline{V}$, can be found in Ref. [12]. In the hypercube, a message is blocked after making i hops if all possible virtual channels at the remaining $(D-i)$ dimensions are busy. The probability, Pb_i, can be written as

$$Pb_i = \left(\sum_{j=1}^{2^L} P_{V,j}\right)^{D-i} \qquad (0 \le i \le D-1) \qquad (13)$$

Let E_i denote the expected time for the header to reach the destination from the current node. If the header succeeds in reserving the required virtual channel and advances to the next node, the residual expected time becomes E_{i+1}. This case occurs with probability $(1-Pb_i)$. On the other hand, if the header encounters blocking and backtracks to the source node, the residual expected time is E_0. Therefore, E_i satisfies the following difference equations [6]

$$E_i = (1-Pb_i)(E_{i+1}+1) + Pb_i(E_0+i) \quad (0 \le i \le D-1) \quad \text{and} \quad E_D = 0 \qquad (14)$$

Solving the above equations yields E_0 as

$$x_i = \begin{cases} 1 & (i=0) \\ \dfrac{x_{i-1} - Pb_i}{1-Pb_{i-1}} & (1 \le i \le D) \end{cases} \quad \text{and} \quad y_i = \begin{cases} 0 & (i=0) \\ \dfrac{x_{i-1} - (i-2)Pb_{i-1} - 1}{1-Pb_{i-1}} & (1 \le i \le D) \end{cases} \qquad (15)$$

$$E_0 = -\frac{y_D}{x_D} \qquad (16)$$

The mean time to set up a path is given by $\overline{E} = E_0 + D$. Due to the requirement of analytic simplicity and practicality, we approximately model the distribution of the path set-up time by an exponential distribution. So $E^*(s)$ can be expressed as [14]

$$E^*(s) = \frac{\alpha}{\alpha + s} \tag{17}$$

where α is selected to match the mean path set-up time and is given by $\alpha = 1/\overline{E}$.

The mean message latency is composed of the mean network latency and the mean waiting time at the source node. However, to model the effect of virtual channel multiplexing, the message latency has to be scaled by the average degree of virtual channel multiplexing that takes place at a given physical channel. Thus, we can write [13]

$$Latency = (\overline{T} + Ws)\overline{V} \tag{18}$$

3 Simulation Experiments

We have developed a discrete-event simulator, operating at the flit level, in order to validate the above analytical model. Each simulation experiment was run until the network converged to its steady state. The cycle time in the simulator is defined as the transmission time of a single flit to cross from one node to the next. Message destinations are uniformly distributed across the network. Figures 1~3 depict results for the mean message latency predicted by the above model plotted against those provided by the simulator in the 4, 6 and 8-dimensional hypercubes, respectively. Message length is *M*=32 and 64 flits. Number of virtual channels of per physical channel is *V*=3, 5 and 7. Hurst parameters are H =0.6, 0.7 and 0.8; Parameter for computing autocorrelation at lag 1 is ρ =0.7, 0.8 and 0.9. We have modelled burstiness over five time scales. The figures reveal that the simulation results closely match those predicted by the analytical model in the steady state region. Its tractability makes it a practical and cost-effective evaluation tool to study the performance behaviour of circuit-switched hypercubes in the presence of multiple time-scale bursty and correlated traffic.

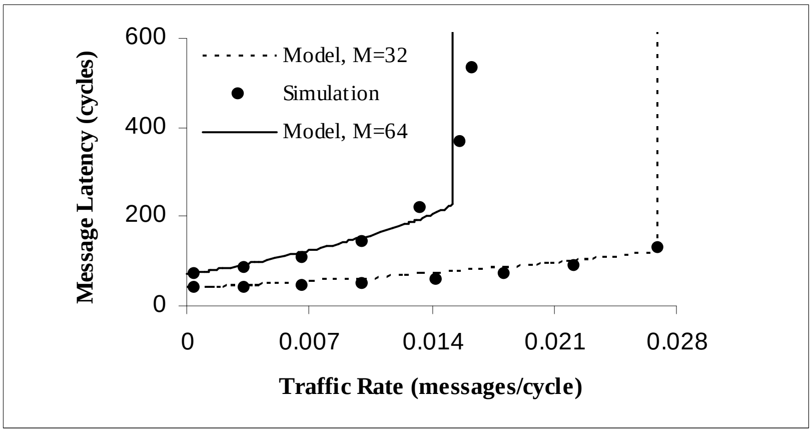

Fig. 1. Latency predicted by the model and simulation in 4-dimensional hypercubes, *V*=3, $H = 0.6, \rho = 0.7$.

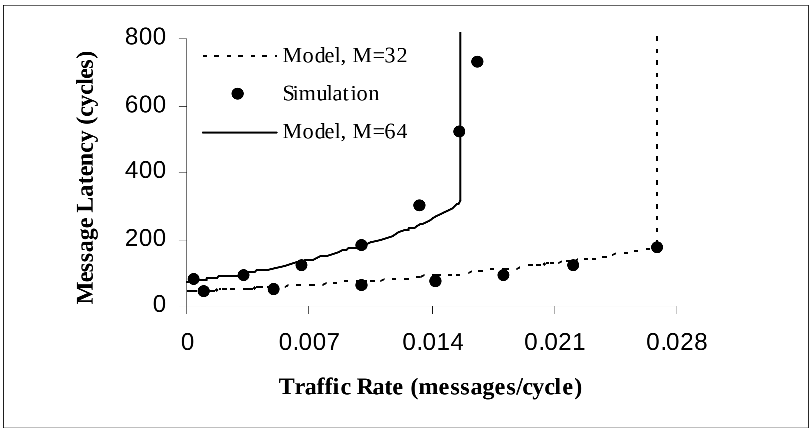

Fig. 2. Latency predicted by the model and simulation in 6-dimensional hypercubes, *V*=5, $H = 0.7, \rho = 0.8$.

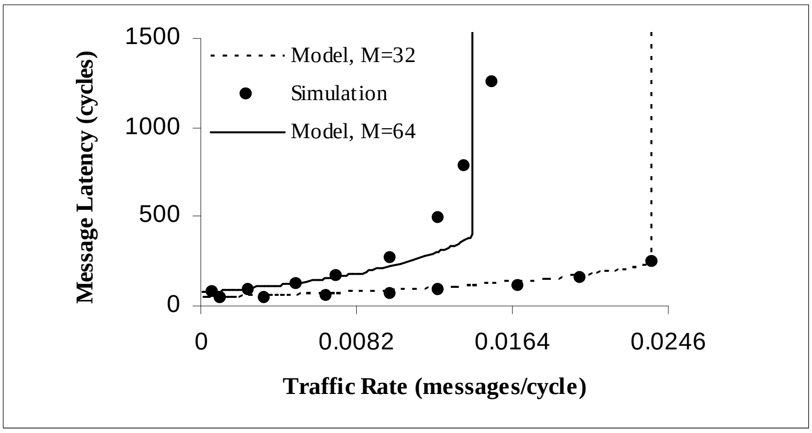

Fig. 3. Latency predicted by the model and simulation in 8-dimensional hypercubes, *V*=7, $H = 0.8, \rho = 0.9$.

4 Conclusions

There has been growing evidence over the past few years that traffic burstiness and correlation over many time scales appear in a variety of systems including local-area and wide-area networks, digitised multimedia systems, web servers, and parallel computation systems. This fractal-like behaviour of traffic exhibits a totally different behaviour from the conventional Poisson process and has great impact on network performance. In an effort towards providing cost-effective tools for hypercube networks, this paper proposes a analytical model for circuit-switched hypercubes in the presence of multiple time-scale bursty and correlated traffic, which is modeled by the by the superposition of a number of different two-state MMPPs. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments of the actual system.

References

1. Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks: An Engineering Approach. IEEE Computer Society Press, Los Alamitos (1997).
2. Loucif S., Ould-Khaoua M., Mackenzie L.M.: Modelling Fully-Adaptive Routing in Hypercubes. Telecommun. Syst. 1 (2000) 111-118.
3. Tanenbaum, A.S.: Computer networks (3rd Edition). Prentice-Hall (1996).
4. Chlamtac, I., Ganz, A., Kienzle, M.G.: A Performance Model of a Connection-Oriented Hypercube Interconnection System. Perform. Eval. 2 (1996) 151-167.
5. Colajanni, M., Ciciani, B., Tucci, S.: Performance Analysis of Circuit-Switching Interconnection Networks with Deterministic and Adaptive Routing. Perform. Eval. 1 (1998) 1-26.
6. Min, G., Ould-Khaoua, M.: Message Latency in Hypercubic Computer Networks with Bursty Traffic Pattern. J. Comput. Electrical Engineering. 3 (2004) 207-222.
7. Sharma, V., Varvarigos, E.A.: Circuit Switching with Input Queuing: An Analysis for the D-Dimensional Wraparound Mesh and the Hypercube. IEEE Trans. Parallel Distrib. Syst. 4 (1997) 349-366.
8. Crovella, M.E., Bestavros, A.: Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. IEEE/ACM Trans. Networking. 6 (1997) 835-846.
9. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the Self-Similar Nature of Ethernet Traffic (Ext. Version). IEEE/ACM Trans. Networking. 1 (1994) 1-15.
10. Sahuquillo, J., Nachiondo, T., Cano, J.C., Gil, J.A., Pont, A.: Self-Similarity in SPLASH-2 Workloads on Shared Memory Multiprocessors Systems. Proc. EURO-PDP’2000. IEEE Computer Society Press. (2000) 293-300.
11. Park, K., Willinger, W. (eds.): Self-Similar Network Traffic and Performance Evaluation. John Wiley & Sons, New York (2000).
12. Min, G., Ould-Khaoua, M.: A Performance Model for K-Ary N-Cube Networks with Self-Similar Traffic. Proc. 16th IEEE & ACM PDPS '2002. IEEE Computer Society Press, Florida (2002) CD-ROM.
13. Ould-Khaoua, M.: A Performance Model for Duato’s Fully Adaptive Routing Algorithm in K-Ary N-Cubes. IEEE Trans. Comput. 12 (1999) 1-8.
14. Kleinrock, L.: Queueing Systems: Theory, Vol. 1. John Wiley & Sons (1975).
15. Andersen, A.T., Nielsen, B.F.: A Markovian Approach for Modeling Packet Traffic with Long-Range Dependence. IEEE J. Selected Areas Commun. 5 (1998) 719-732.
16. Manjunath, D., Sikdar, B.: Input Queued Switches for Variable Length Packets: Analysis for Poisson and Self-Similar Traffic. Comput. Commun. 6 (2002) 590-610.
17. Neuts, M.F., Li, J.: The Bernoulli Splitting of a Markovian Arrival Process. http://www.maths.adelaide.edu.au/jli/papers/split.pdf. (2002).
18. Fischer, W., Meier-Hellstern, K.: The Markov-modulated Poisson process (MMPP) Cookbook. Perform. Eval. 2 (1993) 149-171.