Analysis of Pipelined Circuit Switching in Cube Networks

Geyong Min and Mohamed Ould-Khaoua

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

Abstract. This paper proposes the first analytical model of pipelined circuit switching (PCS) in cube networks. The model uses a Markov chain to analyse the backtracking actions of the header flit during the path set-up phase in PCS. One of the main features of the present model is its ability to capture the effects of using virtual channels. The validity of the model is demonstrated by comparing analytical results to those obtained through simulation experiments.

1 Introduction

Several recent studies have revealed that pipelined circuit switching (PCS) can provide superior performance characteristics over wormhole switching because it combines the advantages of both circuit switching and wormhole switching [2], [4]. In PCS, a reserved path from the source to the destination is set up prior to the transmission of the data as in circuit switching. However, PCS differs in the way that paths are established. When the header cannot progress because all the required *virtual channels* are busy, it releases the last reserved virtual channel by backtracking to the preceding node, then continues its search from the node to find an alternative path to the destination. Since seized channels are released when blocking occurs, deadlock cannot emerge during message routing in PCS. Thus, unlike in wormhole switching, fully adaptive routing can be cheaply implemented in PCS.

This paper presents the first analytical model of PCS in hypercubes (or cubes for short). The model uses a Markov chain to calculate the mean time to set up a path, and M/G/1 queueing systems to compute the mean waiting time that a message experiences at a source before entering the network. Results from simulation show close agreement with those predicted by the model.

2 Analysis

The model is based on the following assumptions: (1) Message destinations are uniformly distributed across the network nodes. (2) Nodes generate traffic independently of each other, which follows a Poisson process with a mean rate of λ_g messages/cycle. (3) Message length is M flits, each of which requires one cycle to cross from one router to the next. (4) 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. (5) Each physical channel is divided into V $(V \geq 1)$ virtual channels. (6) Exhaustive Profitable Backtracking (EPB) [4] routing protocol is used.

A. Bode et al. (Eds.): Euro-Par 2000, LNCS 1900, pp. 904-908, 2000.

The mean message latency is composed of the mean network latency, S, that is the time to cross the network, and the mean waiting time seen by the message in the source node, W_s. However, to model the effects of virtual channels time-multiplexing, the mean message latency has to be scaled by a factor, $\overline{V}$, representing the average degree of virtual channels multiplexing at a given physical channel. Therefore, we can write [6]

$$Latency = (S + W_s)\overline{V} \tag{1}$$

Under the uniform traffic pattern, the message whose destination is i $(1 \le i \le n)$ hops away, can reach $\binom{n}{i}$ nodes out of a total of $(N-1)$ nodes in the network. The average number of channels, d, that a message visits to reach its destination is therefore given by

$$d = \frac{\sum_{i=1}^{n} i \binom{n}{i}}{N-1} = \frac{n}{2}\frac{N}{N-1} \tag{2}$$

The mean network latency, S, consists of two parts: the mean time to set up a path, C, and the actual message transmission time. Thus, S can be written as

$$S = C + d + M \tag{3}$$

In order to calculate the mean path set-up time, C, we use a Markov chain [3] to model the header actions to establish the path. State π_i $(0 \le i \le d)$ in the Markov chain corresponds to the case where the header is at the intermediate node that is i hops away from the source node. Let C_i denote the expected duration to reach state π_d starting from state π_i. A transition out of state π_i to π_{i-1} implies that the header has encountered blocking and has to backtrack to the preceding node. The residual duration becomes C_{i-1}. The transition rate is the probability, P_{b_i}, that the header is blocked in the node corresponding to state π_i. However, the transition out of state π_i to π_{i+1} denotes that the header succeeds in reserving the required virtual channel and advances one hop closer to its destination. The remaining duration is C_{i+1}. The transition rate is the probability $1 - P_{b_i}$. Given that the header requires one cycle to move from one node to the next, the above argument reveals that the expected duration C_i satisfies the difference equations

$$C_i = \left(1 - P_{b_i}\right)C_{i+1} + P_{b_i} C_{i-1} + 1 \qquad (1 \le i \le d\text{-}1) \tag{4}$$

where the states π_0 and π_d satisfy the following boundary conditions

$$C_0 = \left(1 - P_{b_0}\right)\left(C_1 + 1\right) + P_{b_0} C_0 \tag{5}$$

$$C_d = 0 \tag{6}$$

Solving the above equations (4~6) yields the expected duration time, C_0, to reach state π_d starting from state π_0. The mean path set-up time can be written as

$$C = C_0 + d \tag{7}$$

where the term d accounts for the d cycles that are required to send the acknowledgement flit back to the source.

On average, C channels are visited before a path is set up. Half of the visits occur to reserve the virtual channels in the direction leading to the destination node and another half take place in the opposite direction using the reserved channels. Since a router has n output channels and the local node generates λ_g messages per cycle, the mean arrival rate on a channel, λ_c, can be approximated

$$\lambda_c = \frac{\lambda_g C}{2n} \tag{8}$$

The probability of blocking, P_{b_i}, depends on the header's current network position. A header is blocked at the intermediate node that is i hops away from the source if all possible virtual channels at the remaining $(d-i)$ dimensions are busy. Let P_V denote the probability that V virtual channels at a given physical channel are busy (P_V is determined below). The probability, P_{b_i}, can be written as

$$P_{b_i} = (P_V)^{d-i} \qquad (0 \le i \le d-1) \tag{9}$$

The probability, P_t $(0 \le t \le V)$, that t virtual channels at a given physical channel are busy, can be determined using a Markovian model [1]. In the steady state, the model yields the following probabilities.

$$q_t = \begin{cases} 1 & t = 0 \\ q_{t-1}\lambda_c S & 0 < t < V \\ q_{t-1}\lambda_c / (1/S - \lambda_c) & t = V \end{cases} \tag{10}$$

$$P_t = \begin{cases} 1/\sum_{l=0}^{V} q_l & t = 0 \\ P_{t-1}\lambda_c S & 0 < t < V \\ P_{t-1}\lambda_c / (1/S - \lambda_c) & t = V \end{cases} \tag{11}$$

The average degree of multiplexing of virtual channels, that takes place at a given physical channel, is given by [1]

$$\overline{V} = \frac{\sum_{t=0}^{V} t^2 P_t}{\sum_{t=0}^{V} t P_t} \tag{12}$$

To determine the mean waiting time, W_s, that a message experiences in the source node before entering the network, the injection channel is treated as an M/G/1 queue [5]. Since a message in the source node can enter the network through any of the V virtual channels, the mean arrival rate on each virtual channel is λ_g / V. Adaptive routing distributes traffic evenly across the network, and the mean service time at each channel is approximated by the mean network latency, S [6]. Since the minimum mean network latency is $M + 3d$ in the absence of blocking, the variance of the service time distribution can be approximated as [6]

$$\sigma^2 = (S - M - 3d)^2 \tag{13}$$

As a result, the mean waiting time becomes

$$W_s = \frac{(\lambda_g / V)S^2\left(1 + (S - M - 3d)^2 / S^2\right)}{2\left(1 - (\lambda_g / V)S\right)} \tag{14}$$

3 Model Validation

The above model has been validated by means of a discrete-event simulator. Fig. 1 depicts mean message latency results predicted by the above model plotted against those provided by the simulator in the 64 and 256 nodes cube networks. The figure reveals that the simulation results closely match those predicted by the analytical model in the steady state regions.

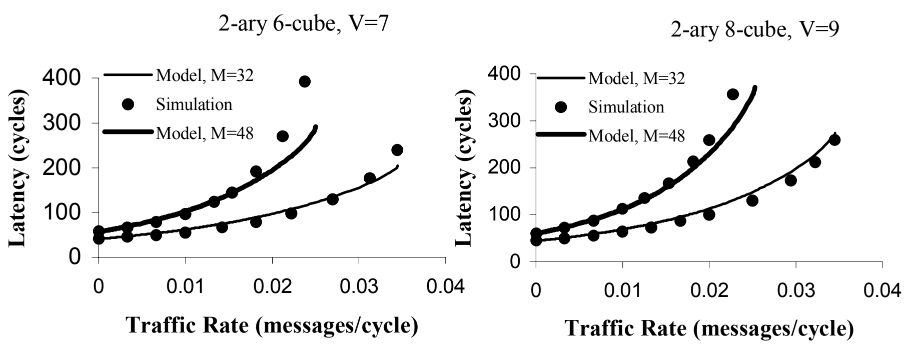

Fig. 1: Latency predicted by the model and simulation, *n*=6 and 8, *V*=7 and 9.

4 Conclusion

This paper has presented an analytical model of PCS in cube networks augmented with virtual channels. The simplicity of the model makes it a practical and cost-effective evaluation tool. The next step in our work is to develop a model for a high-radix k-ary n-cube.

References

1. Dally, W.J.: Virtual channel flow control. IEEE Trans. Parallel & Distributed System. 2 (1992), 194-205
2. Duato, J., Yalamanchili, S., Ni, L.: Interconnection networks: An engineering approach. IEEE Computer Society Press (1997)
3. Feller, W.: An introduction to probability theory and its applications, Vol. 1, John Wiley, New York, (1967)

4. Gaughan, P.T., Yalamanchili, S.: A family of fault-tolerant routing protocols for direct multiprocessor networks. IEEE Trans. Parallel & Distributed System. 5 (1995), 482-497
5. Kleinrock, L.: Queueing Systems Vol. 1, John Wiley, New York (1975)
6. Ould-Khaoua, M.: A performance model for Duato's fully adaptive routing algorithm in k-ary n-cubes. IEEE Trans. Computers. 12 (1999), 1-8