COMMUNICATION DELAY IN WORMHOLE-ROUTED TORUS NETWORKS

A. Shahrabi, M. Ould-Khaoua, L. Mackenzie
Computing Science Department, Glasgow University, Glasgow, UK
Tel: +44 141 339 8855 ext. 0914, Fax: +44 141 330 4913
Email: {alireza, mohamed, lewis}@dcs.gla.ac.uk

ABSTRACT

A new analytical model for predicting message delay in wormhole-routed torus is presented. Unlike previous wormhole routing models, which mainly have been developed for uniform traffic, the model introduced in this paper computes message latency in the wormhole-routed torus in the presence of broadcast traffic. Results obtained through simulation experiments show that the model exhibits a good degree of accuracy in predicting message latency under different working conditions.

Keywords

Interconnection Networks, Wormhole Routing, Adaptive Routing, Broadcast Operation, Performance Modelling.

1. INTRODUCTION

Analytical models of wormhole-routed networks have been widely reported in the literature, e.g. [3], [4], [6], [11]. However, all these models have been discussed in the context of unicast communication. On the other hand, previous research studies of collective communication have focused primarily on the design of efficient algorithms for wormhole-routed networks [10], [12], and there has been comparatively little activity in the area of analytical modelling of these algorithms. As a result, most such studies [8], [10], [12] have relied solely on software simulation to evaluate the performance merits of collective communication. The significant advantage of the analytical approach over simulation is that the analytical models can be used to obtain performance results for large systems that are infeasible by simulation due to the excessive computation demands on conventional computers.

This paper presents a new analytical model to compute message latency in the wormhole-routed torus in the presence of broadcast communication. The broadcast algorithm considered in this study is based on the algorithm proposed by Bose et al [2] for the multiple-port k-ary n-cube. The authors in [2] have shown that this algorithm produces an optimal spanning tree. To illustrate the development of the model, we use Duato's algorithm [7] to route both broadcast and unicast messages as it provides better performance than deterministic routing while maintaining comparable implementation cost [7], [8]. The rest of the paper is organised as follows. Section 2 describes the analytical model while Section 3 validates the model through simulation. Finally, Section 4 concludes the study.

SAC 2002, Madrid, Spain
© 2002 ACM 1-58113-445-2/02/03...$5.00

2. THE ANALYTICAL MODEL

Details of the router structure and broadcast algorithm used in the present model can be found in [13]. The proposed model is based on the following assumptions, which are commonly accepted in the literature [1], [3], [11].

a) Nodes generate traffic independently of each other, via a Poisson process with a mean rate of λg messages per cycle.
b) When a message is generated in a given source node, it has a finite probability β of being a broadcast message and probability $(1-\beta)$ of being unicast. A broadcast message is delivered to every node using the broadcast algorithm. A unicast message is sent to other nodes in the network with equal probability.
c) All messages experience a start-up latency of Δ cycles.
d) Message length is M flits, each of which is transmitted in one cycle across the physical channel.
e) A local queue in a given source node has infinite capacity. Moreover, messages are transferred to the local PE as soon as they arrive at their destinations.
f) V (V>2) virtual channels are used per physical channel. According to Duato's adaptive routing algorithm [4], class a contains V-2 virtual channels, which are crossed adaptively, and class b contains two virtual channel, which are crossed deterministically.

The mean latency of a unicast message, $\overline{L}_u$, is composed of the mean network latency, $\overline{S}_u$, and the mean waiting time seen by a message in the source node, $\overline{W}_s$, before entering the network. However, to model the effects of virtual channel multiplexing the mean message latency has to be scaled by a factor, $\overline{V}$, representing the average degree of virtual channels multiplexing, that takes place at a given physical channel. Therefore, we can write $\overline{L}_u$ as

$$\overline{L}_u = (\overline{S}_u + \overline{W}_s)\,\overline{V} \qquad (1)$$

Before describing how to determine the quantities $\overline{S}_u$, $\overline{W}_s$, and $\overline{V}$, we determine first the traffic rate on a given network channel, λ_c.

Calculation of λ_c : Adaptive routing distributes traffic evenly across network channels as it enables messages to use channels in any order that bring them closer to their destinations. Moreover, given that the destinations for unicast messages are uniformly distributed and the broadcast traffic is balanced across the network channels result in channels having an equal traffic rate. In the broadcast algorithm, a broadcast message is replicated at various stages in the broadcast tree. A replicated message is copied into the local queue of the node to be injected later across the required output channels. So, a source node generates three different type of messages: unicast messages with a rate of $\lambda_{s_u} = (1-\beta)\lambda_g$, one-step broadcast messages with a rate of $\lambda_{s_b} = \beta\lambda_g$, and replicated messages with a rate of λ_{s_r}, which is determined as follows. Given that a source node has generated a broadcast message, the probability that a particular node in the network, other than the source node, replicates the broadcast message and delivers copies to its neighbouring nodes is $\sum_{i=1}^{2n} N_r^i / (k^2-1)$. N_r^i is the number of nodes in the broadcast tree of a torus of radix k that replicate the broadcast message i times and is given by [13]

$$N_r^i = \begin{cases} 2k & i=0 \\ k^2-3k & i=1 \\ 2 & i=2 \\ k-3 & i=3 \end{cases} \quad (2)$$

Since there are (k^2-1) other nodes in the network and the generation rate of broadcast messages is $\lambda_{s_b} = \beta\lambda_g$, the rate of replicated messages originating from a given node is given by

$$\lambda_{s_r} = \sum_{i=1}^{2n} N_r^i \ \lambda_{s_b} = \sum_{i=1}^{2n} N_r^i \ \beta\lambda_g \quad (3)$$

Consider now an output channel. The traffic rate, λ_c, on the channel consists of the rates due to unicast, λ_{c_u}, broadcast, λ_{c_b}, and replicated messages, λ_{c_r}. Thus,

$$\lambda_c = \lambda_{c_u} + \lambda_{c_b} + \lambda_{c_r} \quad (4)$$

In a bi-directional torus, the average numbers of hops that a message makes along a given dimension and across the network, $\bar{k}$ and $\bar{d}$, are given by

$$\bar{k} = \begin{cases} \frac{k}{4} & k \text{ is even} \\ \frac{1}{4}(k - \frac{1}{k}) & k \text{ is odd} \end{cases} \quad (5)$$

$$\bar{d} = n\bar{k} \quad (6)$$

Since a router in the torus has 2n output channels and a node generates, on average, $\lambda_{s_u} = (1-\beta)\lambda_g$ unicast messages in a cycle, the traffic rate of unicast messages, λ_{c_u}, received by each channel in the network is simply

$$\lambda_{c_u} = \frac{(1-\beta)\lambda_g \bar{d}}{2n} \quad (7)$$

A source node generates broadcast messages with a rate $\lambda_{s_b} = \beta\lambda_g$. Since a copy of the broadcast message has to be sent to the all neighbouring nodes through the output channels, the rate of one-step broadcast traffic on a given channel can be expressed as

$$\lambda_{c_b} = \beta\lambda_g \quad (8)$$

In order to compute the traffic rate due to replicated broadcast messages, λ_{c_r}, we need to know the mean number of replications that a given node performs in a given broadcast operation. The number of replication varies from one node to another depending on the node position in the broadcast tree, as shown in Figures 3 and 4. The probability that a broadcast message is replicated i times $(0 \le i \le 3)$ when it reaches an intermediate node is given by

$$P_{r_i} = \frac{N_r^i}{k^2-1} \quad (9)$$

Hence, the mean number of replication of a broadcast message in a given node can be expressed as

$$\bar{\omega} = \sum_{i=0}^{2n} i \ P_{r_i} = \sum_{i=0}^{2n} i \frac{N_r^i}{k^2-1} \quad (10)$$

Given that a replicated message can be sent over one of the output channels with equal probability, the traffic rate of replicated messages on each channel is given by

$$\lambda_{c_r} = \frac{\bar{\omega}}{2n}\lambda_{s_r} = \frac{\bar{\omega}}{4}\sum_{i=1}^{2n} N_r^i \beta\lambda_g \quad (11)$$

Calculation of $\bar{S}_u$: The mean network latency of a unicast message, $\bar{S}_u$, consists of two parts: one is the delay due to the actual message transmission time, and the other is due to blocking in the network. Let B_j the mean blocking time seen by a unicast message at the j-th hop channel $(1 \le j \le \bar{d})$ along its network path. Given that a message makes, on average, $\bar{d}$ hops to reach its destination, $\bar{S}_u$ can be written as

$$\bar{S}_u = M + \bar{d} + \sum_{j=1}^{\bar{d}} B_j \quad (12)$$

where M is the message length. The number of alternate routes that a unicast message can select to advance towards its destination depends on the number of hops already made in both dimensions to reach its current node. When a message arrives at the j-th channel $(1 \le j \le \bar{d})$, it has already made $(j-1)$ hops. These hops can be a combination of (x, y) hops, with x and y being the number of hops achieved in the first and second dimensions respectively, $(x+y=j-1)$ $(0 \le x, y < \bar{k})$. To determine the probability that a message has crossed all channels of one dimension, two cases need to be considered.

a) When $(1 \le j \le \bar{k})$, the number of (x, y) combinations is j. In this case, a message still has to cross channels in both dimensions and, therefore, can choose among adaptive virtual channels of both dimensions.

b) When $(\bar{k} < j \le \bar{d})$, the number of (x, y) combinations is $(\bar{d} - j + 2)$. In only two cases, $(\bar{k}, j-1-\bar{k})$ and $(j-1-\bar{k}, \bar{k})$, out of these combinations, a message has crossed all channels of one dimension, and thus all the remaining hops are to be made on the other dimension.

So, when a message arrives at the j-th channel, the probability that

there remains only one dimension to be crossed, P_{c_j}, can be written as

$$P_{c_j} = \frac{2}{\bar{d} - j + 2} \tag{13}$$

Hence the probability that a message in its next hop can choose any adaptive virtual channel of the two dimensions is $(1 - P_{c_j})$.

A message is blocked at the j-th channel when all the adaptive virtual channels of the remaining dimensions to be visited and also the deterministic virtual channels of the lowest dimension to be visited are busy. When blocking occurs, a message has to wait for the deterministic virtual channel at the lowest dimension. The mean blocking time is a function of the probability of blocking, P_{b_j}, and the mean waiting time, $\overline{W}_c$, for a message to acquire the deterministic channel at the lowest dimension. The mean blocking time can therefore be written as

$$B_j = P_{b_j} \overline{W}_c \tag{14}$$

To compute P_{b_j} we need to compute firstly the probability that all adaptive virtual channels at a dimension are busy, P_a, and secondly the probability that all the adaptive and deterministic virtual channels at a dimension are busy, P_d. To compute P_a, three cases are considered.

1) V virtual channels are busy. This implies that all adaptive virtual channels are busy.
2) $(V-1)$ virtual channels are busy. The number of combinations where $(V-1)$ out of V virtual channels are busy is $\binom{V}{V-1}$. Only two combinations out of $\binom{V}{V-1}$ result in all adaptive virtual channels being busy.
3) $(V-2)$ virtual channels are busy. The number of combinations where $(V-2)$ out of V virtual channels are busy is $\binom{V}{V-2}$. Only one combination out of these results in all adaptive virtual channels being busy.

Similarly, to obtain the second probability, P_d, two cases are considered.

1) V virtual channels are busy. This means that all adaptive and the required deterministic virtual channels are busy.
2) $(V-1)$ virtual channels are busy. In this case only two combinations out of $\binom{V}{V-1}$ result in all adaptive and the deterministic virtual channels being busy.

Let P_v be the probability that v virtual channels at a given physical channel are busy (P_v is computed below). Given that each physical channel is split into V virtual channels and taking into account the different cases mentioned above, P_a and P_d are found to be

$$P_a = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} + \frac{P_{V-2}}{\binom{V}{V-2}} \tag{15}$$

$$P_d = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} \tag{16}$$

Combining equations 13 to 16 yield the probability of blocking, P_{b_j}, at the j-th channel as

$$P_{b_j} = \begin{cases} P_a P_d & 1 \le j \le \bar{k} \\ (1 - P_{c_j}) P_a P_d + P_{c_j} P_d & \bar{k}+1 \le j \le \bar{d} \end{cases} \tag{17}$$

To determine the mean waiting time to acquire a virtual channel, $\overline{W}_c$, in the event of blocking, a physical channel is treated as an M/G/1 queue with a mean waiting time of [9]

$$\overline{W}_c = \frac{\rho \overline{S}(1 + C_{\overline{S}}^2)}{2(1-\rho)} \tag{18}$$

$$\rho = \lambda_c \overline{S} \tag{19}$$

$$C_{\overline{S}}^2 = \frac{\sigma_{\overline{S}}^2}{\overline{S}^2} \tag{20}$$

where λ_c is the traffic rate on a network channel, $\overline{S}$ is the mean service time, and $\sigma_{\overline{S}}^2$ is the variance of the service time distribution. While the traffic rate, λ_c, is given by equation 4, the other two quantities, $\overline{S}$ and $\sigma_{\overline{S}}^2$ are computed as follows. One-step broadcast and unicast messages see different network latencies time as they cross a different number of channels to reach their destinations. A unicast message sees a mean network latency, $\overline{S}_u$, given by equation 12, whereas a one-step broadcast message sees a mean network latency $\overline{S}_b$. The mean service time seen by an arbitrary message considering broadcast and unicast messages with thier appropriate weights is given by

$$\overline{S} = \frac{\lambda_{c_b} + \lambda_{c_r}}{\lambda_c} \overline{S}_b + \frac{\lambda_{c_u}}{\lambda_c} \overline{S}_u \tag{21}$$

To ease the development of our model while maintain a good degree of accuracy in predicting message latency we follow a suggestion of Draper and Ghosh [6] for computing the variance of the service time. Since the minimum service time at a channel is equal to the message length, the variance of the service time distribution can be approximated as

$$\sigma_{\overline{S}}^2 = (\overline{S} - M)^2 \tag{22}$$

As a result, the mean waiting time becomes

$$\overline{W}_c = \frac{\lambda_c \overline{S}^2 \left(1 + \frac{(\overline{S} - M)^2}{\overline{S}^2}\right)}{2(1 - \lambda_c \overline{S})} \tag{23}$$

The mean network latency of a one-step broadcast message, $\overline{S}_b$, is determined in a similar manner to the case of unicast message. Since a one-step broadcast message makes one hop to reach the next destination node, $\overline{S}_b$ can be written as

$$\overline{S}_b = M + B_b \tag{24}$$

A one-step broadcast message can use only one specific output channel to reach its destination. As a result, the message suffers from blocking when all the adaptive virtual channels and the deterministic virtual channel belonging to the output channel are

busy. Since there is a balance traffic on network channels, the message sees the same mean waiting time, $\overline{W}_c$, to acquire a virtual channel at an output channel, regardless of its position in the network. Given that a one-step broadcast message is blocked when all the V virtual channel at the required output channel are busy, the mean blocking time, B_b, can be written as

$$B_b = P_V \overline{W}_c \tag{25}$$

The above equations reveal that there exist several interdependencies between the different variables of the model. For instance, equation 21 shows that $\overline{S}$ is a function of $\overline{S}_u$ and $\overline{S}_b$ while equation 12 and 24 show that $\overline{S}_u$ and $\overline{S}_b$ are functions of $\overline{S}$. Since obtaining closed-form expressions for such interdependencies is generally difficult, the different variable of the model are computed using iterative techniques for solving equations [3].

Calculation of $\overline{W}_s$: The mean waiting time in the source node is calculated in a similar manner to that for a network channel (equations 18 to 20). By modelling the injection channel in the source node as an M/G/1 queue, the mean arrival rate and mean service time are given by the following equations

$$\lambda_s = \frac{\lambda_{S_u}}{2n} + \lambda_{S_b} + \frac{\overline{\omega}}{2n}\lambda_{S_r} \tag{26}$$

$$\overline{S}_s = \frac{\lambda_{s_b} + \lambda_{s_r}}{\lambda_{s_u} + \lambda_{s_b} + \lambda_{s_r}}\overline{S}_b + \frac{\lambda_{s_u}}{\lambda_{s_u} + \lambda_{s_b} + \lambda_{s_r}}\overline{S}_u \tag{27}$$

Approximating the variance of the service time distribution by $(\overline{S}_s - M)^2$ yields a mean waiting time at the source as

$$\overline{W}_s = \frac{\lambda_s \overline{S}_s^2 \left(1 + \frac{(\overline{S}_s - M)^2}{\overline{S}_s^2}\right)}{2(1 - \lambda_s \overline{S}_s)} \tag{28}$$

Calculation of $\overline{V}$: The probability, P_v, that v adaptive virtual channels are busy in a physical channel can be determined using a Markovian model [5]. State π_v corresponds to v virtual channels being busy. The transition rate out of state π_i to π_{i+1} is λ_c, where λ_c is the traffic rate on a network channel (and is given by equation 4), while the rate out of π_i to π_{i-1} is $1/\overline{S}$. The transition rate out of the last state, π_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.

$$q_v = \begin{cases} 1 & v = 0 \\ q_{v-1}\lambda_c \overline{S} & 0 < v < V \\ q_{v-1}\dfrac{\lambda_c}{1/\overline{S} - \lambda_c} & v = V \end{cases} \tag{29}$$

$$P_v = \begin{cases} \left(\sum_{j=0}^{V} q_j\right)^{-1} & v = 0 \\ P_{v-1}\lambda_c \overline{S} & 0 < v < V \\ P_{v-1}\dfrac{\lambda_c}{1/\overline{S} - \lambda_c} & v = V \end{cases} \tag{30}$$

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, which takes place at a given physical channel, is given by [5]

$$\overline{V} = \frac{\sum_{i=1}^{V} i^2 P_i}{\sum_{i=1}^{V} i P_i} \tag{31}$$

3. MODEL VALIDATION

The above model has been validated using a discrete-event simulator that performs a time-step simulation of network operations at the flit level. Each simulation experiment is run until the network reaches its steady state; that is until a further increase in simulated network cycles does not change the collected statistics appreciably. Statistics gathering was inhibited for the first 20000 unicast messages to avoid distortions due to the startup transient. Extensive validation experiments have been performed for several combinations of network sizes, message lengths, different fractions of broadcast messages and virtual channels. For the sake of specific illustration, latency results are presented for the networks with $N = 8 \times 8$, $N = 10 \times 10$ and $N = 16 \times 16$ nodes, V=3, 4 and 5 virtual channels per physical channel, message length M=16, 32, 48 and 64 flits and broadcast portion $\beta = 0.02$ and 0.04.

Figure 1 depicts results from the mean unicast message latency predicted by the above analytical model plotted against those provided by the simulator as a function of traffic injection for $N = 8 \times 8$, $N = 10 \times 10$, $N = 16 \times 16$. The horizontal axis in the figure represent the message generation rate of every node per cycle, λ_g, while the vertical axis shows the unicast message latency. The figures reveal that the simulation results closely match those predicted by the analytical model in the steady state regions (i.e. under light and moderate traffic) and even when the network starts to approach saturation. However, the discrepancies in the results near saturation are noticeable. This is due to the approximations which have been made to simplify the development of the model, such as that made in equation 22 for determining the variance of service time at a network channel; this approximation greatly simplifies the model as it allows us to avoid computing the exact distribution of the message service time at a given channel, and which is not a straightforward task due to the interdependencies between service times at successive channels as wormhole routing relies on a blocking mechanism for flow

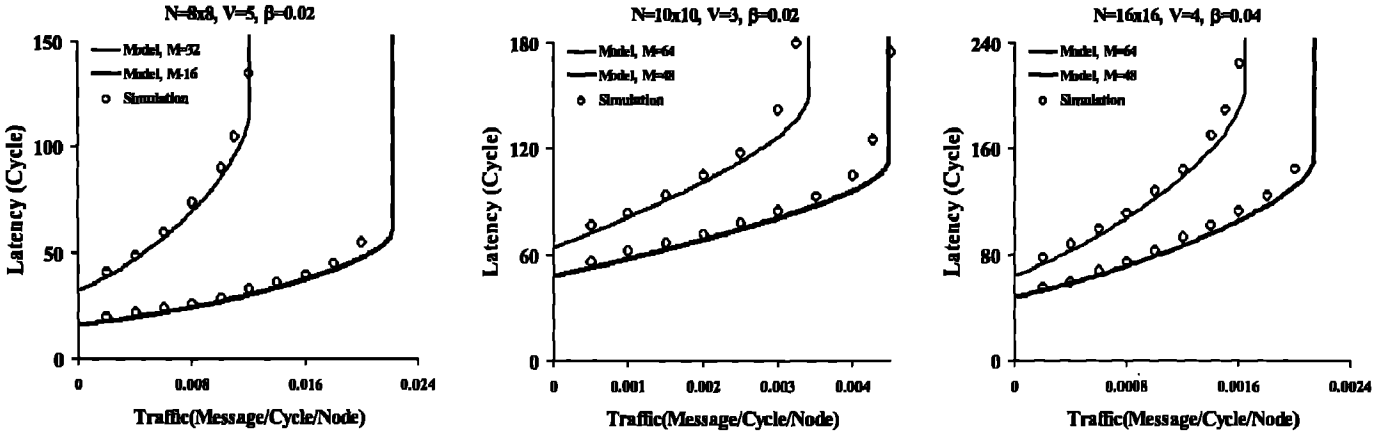

Figure 1: Validation of the unicast latency predicted by the model against simulation in the torus with k=8, 10 and 16, Message length M = 16, 32, 48 and 64, Broadcast portion β = 0.02 and 0.04, and number of virtual channels V = 3, 4, and 5.

control. Nevertheless, we can conclude that the model produces accurate results in the steady state regions, and its simplicity makes it a practical evaluation tool that can be used to gain insight into the behaviour of wormhole-routed torus in the presence of broadcast communication.

4. CONCLUSION

Although many broadcast algorithms have been proposed for common multicomputer networks, e.g. tori, over the past decade there has been little development of analytical models of these algorithms. This paper has presented an analytical model capable of computing unicast latency in wormhole-routed tori under a number of reasonable assumptions. Extensive simulation experiments have shown that the analytical model predicts latency with a good degree of accuracy under different traffic conditions. An obvious continuation of this work would extend the present model to other common multicomputer networks such as n-dimensional meshes. Another line of progression would be to develop new analytical models for the recently proposed multi-destination-based broadcast algorithms, such as those based on the Base Routing Conformed Path (BRCP) methodology [12].

5. REFERENCES

[1] S. Abraham, K. Padmanabhan, Performance of the direct binary *n*-cube network for multiprocessors, *IEEE Transaction on Computers* 38(7) (1989) 1001-1011.

[2] B. Bose, et al, Lee distance and Topological Properties of k-ary n-cubes, *IEEE Transaction on Computers* 44(8) (1995) 1021-1030.

[3] Y. Boura, C.R. Das, T.M. Jacob, A performance model for adaptive routing in hypercubes, Proceedings of International Workshop Parallel processing (1994) 11-16.

[4] W.J. Dally, Performance analysis of k-ary n-cubes interconnection networks, *IEEE Transaction on Computers* 39(6) (1990) 775-785.

[5] W. J. Dally, Virtual channel flow control, *IEEE Transaction on Parallel & Distributed Systems* 3(2) (1992) 194-205.

[6] J.T. Draper, J. Ghosh, A comprehensive analytical model for wormhole routing in multicomputer systems, *Journal of Parallel & Distributed Computing* (32) (1994) 202-214.

[7] J. Duato, A new theory of deadlock-free adaptive routing in wormhole routing networks, *IEEE Transaction on Parallel & Distributed Systems* 4(12) (1993) 320-1331.

[8] J. Duato, S. Yalamanchili, L. Ni, Interconnection networks: An engineering approach (*IEEE Computer Society Press*, 1997).

[9] L. Kleinrock, Queueing Systems (1) (John Wiley, New York, 1975).

[10] P. K. McKinley, H. Xu, A. H. Esfahanian, and L. M. Ni, Unicast-based multicast communication in wormhole-routed networks, *IEEE Transaction on Parallel and Distributed Systems* 5(12) (1994) 1252-1265.

[11] M. Ould-Khaoua, A performance model for Duato's fully-adaptive routing algorithm in k-ary n-cubes, *IEEE Transaction on Computers* 42(12) (1999) 1-8.

[12] D. Panda, S. Singal, R. Kesavan, Multidestination message passing in wormhole k-ary n-cube networks with base routing conformed paths, *IEEE Transaction on Parallel & Distributed Systems* 10(1) (1999) 76-96.

[13] A. Shahrabi, M. Ould-Khaoua, L. Mackenzie, Analytical modeling of broadcast in Torus Networks,, *Tech. Report*, Department of Computing Science, University of Glasgow, 2001.