Deadlock Probability in Unrestricted Wormhole Routing Networks

Arne Folkestad
Norwegian University of Science and Technology
Dept. of Telematics
N-7034 Trondheim, Norway
Arne.Folkestad@item.ntnu.no

Christian Roche
UCLA Computer Science Dept.
3732H Boelter Hall
Los Angeles, CA 90095, USA
roche@cs.ucla.edu

Abstract

To minimize latency in local area networks, the wormhole routing *technique can be used. Drawbacks of this technique are the blocking of messages and, if backpressure flow control is used, the possibility of deadlock. Two principal mechanisms can be employed to deal with deadlocks, namely a deadlock-free routing scheme or a deadlock resolution mechanism. The relative performance of these two mechanisms depends mainly on the deadlock frequency. In this paper, we study analytically the probability of deadlock in an unrestricted wormhole routing network. We capture the topology of the system in two parameters with intuitive meaning, and we derive a closed form expression for an upper bound of the deadlock probability. Simulation runs with different topologies provide good confirmation for the analytical results.*

1 Introduction

Wormhole routing has been successfully used in interconnection networks in multiprocessor computers. Recently, LANs using this technique have become commercially available. One such effort is the 640 Mbps *Myrinet* [1], which is the LAN technology chosen for the Supercomputer Supernet [2], currently being developed at UCLA, JPL and Aerospace. Key characteristics of a wormhole routing network are: uni- or bi-directional point-to-point links, multihop paths, variable length messages, wormhole switching, backpressure flow control, and possibly source routing [3].

Wormhole routing is a technique where a switch immediately forwards an incoming message to the desired output link if this link is available. The forwarding is done before the whole message is assembled in the switch. Hence, a message may stretch across several links and switches, setting up a circuit-like connection for the message transfer. If the desired links are available as the message advances on its path, buffering in intermediate switches is avoided. The message transfer time then reduces to the path propagation delay.

When a message's desired output link is used by another message, a *collision* takes place, and the (incoming) message is blocked. A portion of the message must be stored in a *slack buffer*. If the message lengths were limited and the slack buffers large enough, the transmission could continue, eventually assembling the whole message in the buffer. This scheme is called *cut-through switching* [4]. However, with wormhole routing the message lengths are not limited and the buffers are small. Hence, a backpressure flow control mechanism must be used. Control signals are sent upstream on the reverse link to stop or resume message transmission. The blocked message retains the resources already acquired.

A drawback of the backpressure mechanism are the message blockings, which increase the message transfer time. Messages may be blocked by a message which itself is blocked, creating second and higher order blockings. The most severe blocking occurs when a cycle of blocked messages has been created. In this case, none of the messages can advance towards its destination because each message holds resources needed by other messages. Hence, the messages are blocked forever. Such a situation is called a *deadlock*.

The number of blockings can be reduced using adaptive routing instead of deterministic routing. However, adaptive routing requires more switch intelligence than is normally provided in simple, low-cost LAN switches. *Source routing* is one deterministic routing scheme which is well suited for a network with simple switches. Here, the source determines the route for the whole path. The path is described by a sequence of output links leading to the destination. The routing information is included in the message header.

There are two main approaches to dealing with deadlocks. One is to use a *deadlock-free* routing scheme. This is done by restricting the available paths in a way guaranteeing that no cycles of blocked messages can be created. However, this may imply unbalanced link load and lead to non-minimal paths. Another approach is to use a deadlock-prone *unrestricted* routing scheme, but then have efficient detection and resolution methods. The drawbacks here are that resources must be used on deadlock detection and that some messages must be dropped to break deadlocks. The relative performance of these two schemes depends on the deadlock frequency.

In this paper, we address the probability of deadlock in wormhole routing networks using an unrestricted deterministic routing scheme. We study the occurrence of deadlocks, and outline a method for deadlock detection and resolution. We derive analytically a closed form expression of the probability that a message is dropped due to deadlock. Simulation runs with different network topologies provide good confirmation for the analytical results.

The paper is organized as follows: Section 2 reviews deadlock-free and unrestricted routing schemes. In Section 3, we present the model for analyzing deadlock. The analysis is carried through in Section 4, while Section 5 compares the analytical results with simulation results. Section 6 concludes the paper, and indicates directions for future research.

2 Deadlock handling

Figure 1 depicts a network with four switches in a ring topology. The links are bidirectional. Four messages have entered the network, one at each switch. Each message has a destination that is on the opposite side of the ring, two hops away. Each message has two possible paths, one clockwise and one counterclockwise. If all four messages choose their counterclockwise path, the deadlocked situation in the figure may result. Here, each message has acquired the link on the first hop, but is blocked on the second-hop link. We see that the sequence of requested links forms a cycle.

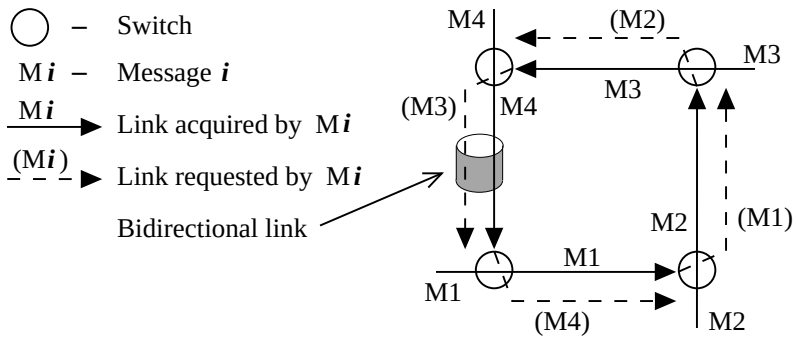

Figure 1: *Four messages have entered the network through different switches. Each message has acquired the first-hop link, but is blocked on the second-hop link. A deadlock results.*

It was early made clear that with deterministic routing, deadlock freedom can only be guaranteed if the routing scheme is acyclic [5]. This means that there is no cycle in the sequence in which links can be requested. One approach to avoiding such cycles is to restrict the paths messages may take in the network. In the figure, we obtain this by restricting the messages entering in one of the switches to use the clockwise path. An example of such a technique is the Up/Down routing scheme [6], which is used in Myrinet.

A major drawback of acyclic routing is that it leads to non-minimal paths and concentration of traffic on some links. Moreover, in the case of multicasting, deadlock-free routing schemes may be difficult to design and demanding in terms of network resources [7]. It may thus be more cost-effective to relax the requirement of deadlock freedom, using efficient deadlock detection and resolution mechanisms instead.

One possible method to detect deadlocks is to start a timer when a message arrives at a switch. If the timer expires, a message that makes no progress is detected. However, in order to distinguish a real deadlock from lack of progress due to non-deadlock blocking, the value of the timer must be large. In the event of deadlock, this locks resources for a long time, causing reduced efficiency. Another possible method is the sending of a control signal on the reverse link of a blocked message. This signal carries information that uniquely identifies the blocking, e.g., the switch and message identifiers. It is forwarded upstream towards the tail of the blocked message. In addition, it is propagated along all messages that are blocked by the message (Figure 2). If a switch receives its own control signal, a cycle of blocked messages, i.e., a deadlock, is detected.

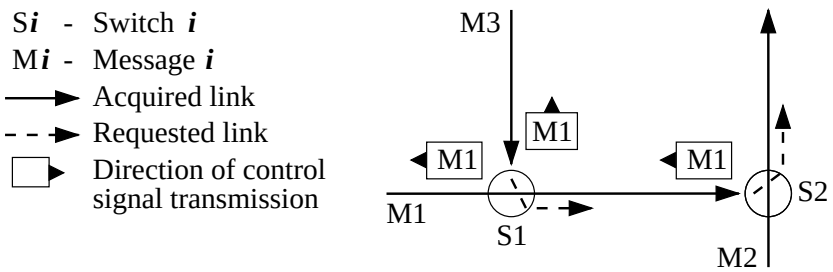

Figure 2: *Message M1 is blocked at switch S2. A control signal is sent upstream to S1, which forwards it towards M1's tail. S1 also sends a copy towards M3's tail.*

Upon deadlock detection, a message must be removed to break the cycle. One approach is to let the switch that receives its own control signal drop the deadlocking message. Thus, the resources held by the message are released, but the message itself is lost.

The relative performance of the two types of routing schemes depends on the resources spent on avoiding or resolving deadlocks, and thus on the deadlock frequency. In the rest of this paper we study analytically the frequency of deadlock in an unrestricted wormhole routing network, using a simplified network model.

3 Model

We consider a network with N unidirectional links with equal capacity C and propagation delay T. The network can have any topology. Note that the backpressure flow control mechanism requires bidirectional links. Here, we assume that the reverse link is limited to carrying control signals only.

To simplify, we consider messages of fixed length M, which cross the network on paths of fixed length L. Both M and L are expressed as a number of links. This means that if b is the message size in bits, we have that $M = b/CT$.

Message arrivals follow a Poisson process with intensity λ. The probability of observing k arrivals in a time interval of length τ is

$$P_k(\tau) = \frac{(\lambda\tau)^k}{k!}\, e^{-\lambda\tau} \tag{1}$$

Now, let m be the average number of messages in the network. We use the convention that a message enters and leaves the network when its head enters and leaves, respectively. If t denotes the average message transfer time, using Little's result, the following relation yields: $m = \lambda t$.

In a network comprising unidirectional links, a deadlock can involve as few as two messages. If $k \geq 2$ messages are simultaneously present in the network, a deadlock involving i, $2 \leq i \leq k$, messages is possible. A deadlock involving exactly i messages is called an *i-deadlock*.

In Figure 3a, we see the two messages M_1 and M_2. They collide at link c_1 as well as link c_2. The collisions have taken place in such an order that the messages block each other. Each message is requesting a link the other message is holding, and a 2-deadlock has been created. In Figure 3b, the three messages M_1, M_2 and M_3 are shown. There is collision at the three links c_1, c_2 and c_3. Each message has acquired one of these links, but is blocked at another. Since each message is requesting a link that one of the other two messages holds a 3-deadlock results. In the two examples, we observe that a deadlock has been created because in a set of messages there is conflicting interest (collisions) regarding the use of certain links. This will be analyzed further in Section 4.

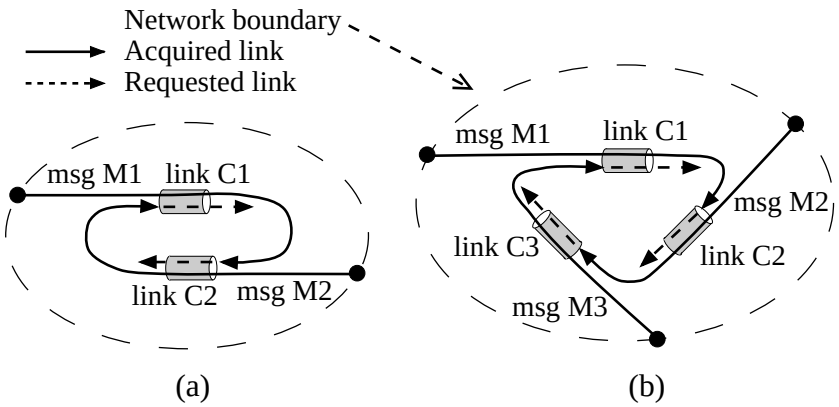

Figure 3: (a) *A 2-deadlock.* (b) *A 3-deadlock.*

To resolve an i-deadlock, we assume that exactly one message is dropped. Its resources are released and can then be acquired by the other messages. Thus, the deadlocked resource request cycle is broken.

When studying the occurrence of deadlocks, we assume that the message length is at least as long as the path length, i.e., $M \geq L$. This assumption allows us to consider the paths only, and not the individual messages. Our abstract network model can now simply be described by the list of available paths and the traffic pattern, i.e., the offered traffic to each path. If some messages were shorter than the traveled path the deadlock probability would be smaller. Hence, compared to a real network with variable length messages, our approach gives a conservative estimate of the deadlock probability.

In this study, focus is on unrestricted wormhole routing schemes. However, the model is valid for any deterministic wormhole routing scheme. The routing scheme will be reflected by the list of available paths.

4 Deadlock analysis

In this section we derive an upper bound of the probability p for a message to be dropped due to a deadlock. First, we express a set of necessary conditions for a deadlock to be created. Then, considering a situation where k messages are simultaneously present in the network, we estimate the probability that an i-deadlock occurs. The probability p is then found by removing the condition on k messages.

We start out by considering k messages in the network, $k \geq 2$. Let $\{M_1, \ldots, M_k\}$ denote the k messages and $\{L_1, \ldots, L_k\}$ their respective paths. Message M_1 is the marked message throughout the analysis.

We assume that a deadlock occurs when the following necessary conditions are fulfilled:

1. Common link condition: There exist an ordering of the paths such that the pairs (L_1, L_2), (L_2, L_3), $\ldots$, (L_k, L_1) each has at least one link in common. Let c_1, c_2, $\ldots$, c_k, respectively, represent the common links.
2. Link order condition: For each path L_j, link c_j appears before (closer to the source than) link c_{j+1}.
3. Timing condition: The k messages arrive within an interval of length $\tau = 2t$.

Common link condition: Possible paths for the messages in Figures 3a and 3b, are depicted in Figures 4a and 5a. The paths can be ordered such that L_j and L_{j+1} have a common link c_j. The common link condition is thus fulfilled.

Link order condition: Part (b) of the figures shows a stretched-out picture of the paths. The link order condition states that there must be an ordering of the common links. This condition is fulfilled here.

Timing condition: Referring to Figure 6, we explain the timing condition. Recall that t is the average message transfer time, defined as the interval between the message head's arrival to and departure from the network. A message can only collide when its head is advancing through the network. When the head reaches the destination, the "circuit" is set up and the rest of the message

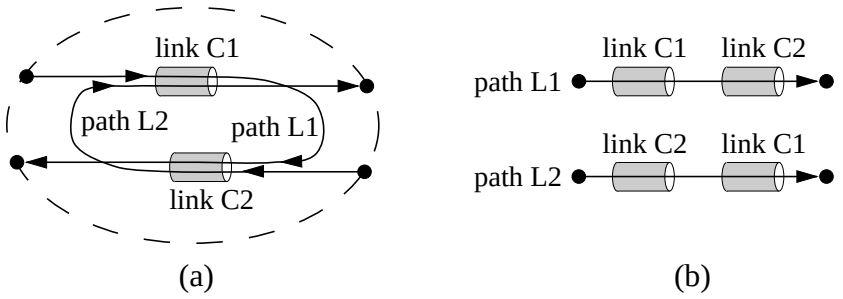

Figure 4: *A pair of paths with two common links.* (a) *The paths in the network.* (b) *Stretched-out picture of the paths.*

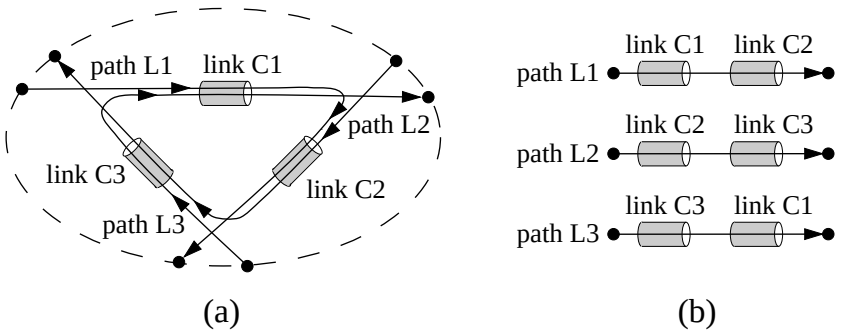

Figure 5: *Three paths, each path pair has one common link.* (a) *The paths in the network.* (b) *Stretched-out picture.*

is transmitted uninterrupted. Let the marked message M_1 arrive at time t_1. M_1 may collide with any message that arrived t time units before itself, or arrives t time units after itself. If a deadlock involving k messages shall be possible, they must thus arrive within an interval of length $\tau = 2t$. This gives the timing condition.

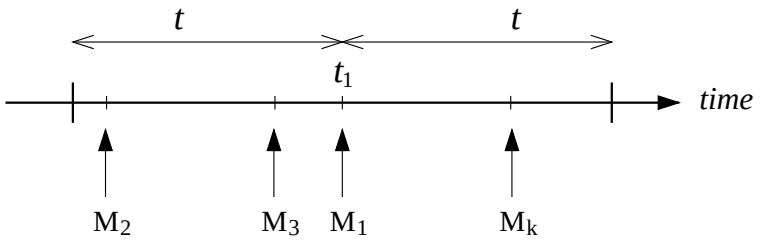

Figure 6: *Typical time interval for message interaction.*

The timing condition is expressed by Eq. (1). The other two conditions, however, have so far no convenient representation. They are related to the network topology and the traffic pattern. To capture this relation, we define the two parameters α and β, as explained below. Their values are found by considering all pairs of possible paths, and the offered traffic to each path in these pairs. In general, α and β cannot be derived analytically but are found numerically.

Parameter α is given a value that is half of the probability that any two possible paths have at least one link in common. The reason for dividing this probability by 2 is to comply with the link order condition. Using α in the case of 2-deadlock gives the value α^2 for the probability of having two links in common. However, for many network topologies the accuracy of this result is

unsatisfactory. To increase the accuracy of the analysis we introduce the parameter β, which gives a more precise expression for the necessary conditions 1 and 2 in the particular case of 2-deadlock. Parameter β is the probability that any two paths have two links in common, and the common links are ordered in a way making a 2-deadlock possible. This means that the pair (L_1, L_2) has links c_1 and c_2 in common. These two links must be ordered such that the link order condition is fulfilled. Figure 4 shows the path pairs we are seeking when computing β.

Let us now consider a situation where k messages enter the network in a time interval of length τ. Conditioning on the k arrivals, let p_k be the probability that the marked message is dropped due to deadlock. Using Eq. (1), the expression for the unconditioned deadlock probability p is

$$p = \frac{\sum_{k=2}^{\infty} p_k P_k(\tau)}{\sum_{k=1}^{\infty} P_k(\tau)} \tag{2}$$

With k messages arriving in interval τ, deadlocks can be created involving i messages, $2 \leq i \leq k$. Let Q_{ki} be the probability that the marked message M_1 is caught in such an i-deadlock. The messages involved in this deadlock are denoted $\{M_1, M_2, \ldots, M_i\}$. Only one (arbitrary) message is dropped when resolving this i-deadlock, the probability that the marked message is the one being dropped is therefore $1/i$. This results in the following expression:

$$p_k = \sum_{i=2}^{k} \frac{Q_{ki}}{i} \tag{3}$$

For $i = 2$, we have $Q_{k2} = (k-1)\beta$. Now let us consider the case $i \geq 3$. There are $\binom{k-1}{i-1}$ possible choices for the $i-1$ messages $\{M_2, \ldots, M_i\}$. The probability of M_2 having a collision with M_1 while not creating a 2-deadlock is $\alpha(1-\gamma)$, where γ is such that $\alpha\gamma = \beta$. The probability of M_3 having a collision with M_2 while not creating a 3-deadlock is $\alpha(1-\alpha)$. The same applies for $M_4, \ldots, M_{i-1}$. Finally, the probability of M_i having a collision with M_{i-1} and creating an i-deadlock is α^2. This reasoning leads us to write the quantities Q_{ki} as

$$\begin{aligned} Q_{k2} &= (k-1)\,\beta, \\ Q_{ki} &= \binom{k-1}{i-1} \alpha^2 (\alpha - \beta)\, \delta^{i-3} \qquad \text{for } i \geq 3 \end{aligned}$$

where $\delta = \alpha(1-\alpha)$. Substituting into Eq. (3) gives $p_2 = \beta/2$ and for $k \geq 3$ (this expression also holds for $k = 2$)

$$\begin{aligned} p_k &= \frac{\beta}{2}(k-1) + \alpha^2(\alpha-\beta) \sum_{i=3}^{k} \binom{k-1}{i-1} \frac{\delta^{i-3}}{i} \\ &= \frac{\beta}{2}(k-1) + \frac{\alpha-\beta}{(1-\alpha)^2} \left[\frac{(1+\delta)^k - 1}{k\delta} - 1 - \frac{k-1}{2}\delta \right] \end{aligned} \tag{4}$$

Substituting into Eq. (2), we derive the following closed form expression of p:

$$p = \frac{1}{2} \frac{\beta - \alpha^2}{1-\alpha} \left(\frac{we^w}{e^w - 1} - 1 \right) + \frac{\alpha - \beta}{(1-\alpha)^2} \left(\frac{1}{e^w - 1} \int_0^w \frac{e^{\delta u} - 1}{\delta u}\, e^u\, du - 1 \right) \tag{5}$$

where $w = \lambda\tau = 2m$. We have now obtained an expression for the deadlock probability p. This expression is a function of the average number m of messages in the system, a quantity which is directly related to the system load. An expression of m as a function of λ is found in [8].

5 Numerical results

We now compare the analytical results with simulations. Four network topologies were chosen: 8-node and 64-node shufflenets, and 9-node and 36-node torus networks, see Figure 7. In all simulations the message length is $M = 10$. The system parameters are presented in Table 1. Note that the value of L equals the longest path in the respective networks. This means that each source sends to the destination that is the furthest away, in terms of (shortest) path length.

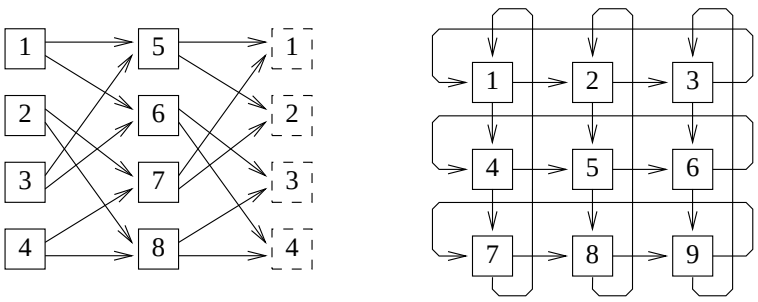

Figure 7: (a) *8-node shufflenet.* (b) *9-node torus network.*

Topology	L	α	β
8-node shufflenet	3	0.2054	0.0179
64-node shufflenet	7	0.1153	0.0065
9-node torus netw.	4	0.2871	0.0352
36-node torus netw.	10	0.3206	0.0947

Table 1: *Parameters in the simulations.*

Figures 8 and 9 compare the simulation and analytical results for the shufflenets and torus networks. The graphs show that the analytical results give a quite good estimate of the measured deadlock probability for small values of m, i.e., at light loads. At higher loads the analytical values are larger than the actual deadlock probability. The discrepancy for higher loads comes partly from the following: Consider two messages on two paths where a 2-deadlock is possible. The messages may get caught in such a deadlock, but may also get caught in a higher-order deadlock. The probability for the latter increases with the number of messages in the network. Thus, the number of lower-order deadlocks is over-evaluated. Therefore, as intended, the estimate gives an upper bound of the deadlock probability.

The torus networks experience more deadlocks than the shufflenets. In Table 1, we see that their values of α and β are greater than those for the shufflenets. An intuitive explanation for this is that the shufflenets have fewer paths that overlap in a way making deadlock possible.

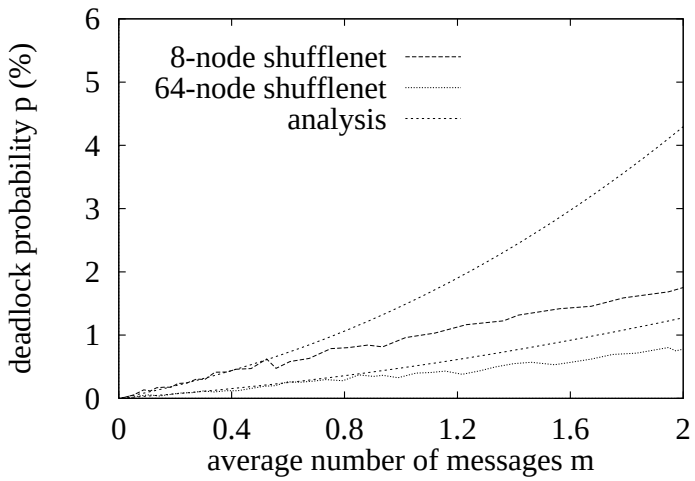

Figure 8: *Deadlock probability p vs. m, case of shufflenets.*

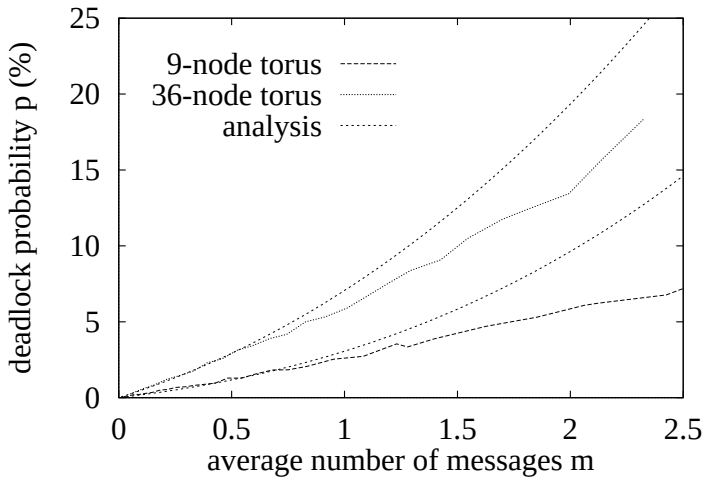

Figure 9: *Deadlock prob. p vs. m, case of torus networks.*

6 Conclusions

In this study we have estimated analytically the probability of deadlock in unrestricted wormhole routing networks with different topologies. We have captured the essential characteristics of the system in a small number of parameters which have simple intuitive interpretation. By expressing a set of necessary conditions for a deadlock situation to occur, we have been able to assess a simple and conservative expression for the deadlock probability. Simulation results provide good confirmation for the analytical results.

This study is interesting since it gives a simple model for the difficult problem of deadlock in wormhole routing networks. It increases the understanding of how the network topology influences the deadlock probability and thus the performance of the system.

Possible directions for future research include systems with bidirectional links and more general traffic patterns. We should also study how much the deadlock frequency decreases if $M < L$. Finally, a comparative study of deadlock-free wormhole routing schemes versus unrestricted wormhole routing schemes should be conducted using the model presented in this paper.

References

[1] N. Boden *et al.*, "Myrinet: A Gigabit-Per-Second Local-Area Network," *IEEE Micro*, vol. 15, no. 1, Feb. 1995.

[2] L. Kleinrock *et al.*, "The Supercomputer Supernet: A Scalable Distributed Terabit Network," *Journal of High Speed Networks*, vol. 4, no. 4, pp. 407–424, 1995.

[3] L. Ni and P. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *IEEE Computer*, vol. 26, no. 2, Feb. 1993.

[4] P. Kermani and L. Kleinrock, "Virtual Cut-Through: A New Computer Communication Switching Technique," *Computer Networks*, vol. 3, no. 4, Sept. 1979.

[5] W. Dally and C. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. Comm.*, vol. C-36, no. 5, May 1987.

[6] M. Schroeder *et al.*, "Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links," *IEEE JSAC*, vol. 9, no. 8, pp. 1318–35, Oct. 1991.

[7] M. Gerla, P. Palnati and S. Walton, "Multicasting Protocols for High-Speed, Wormhole-Routing Local Area Networks," *ACM SIGCOMM'96*, Palo Alto, CA, Aug. 1996.

[8] C. Roche, P. Palnati and M. Gerla, "Performance of Congestion Control Mechanisms in Wormhole Routing Networks," *IEEE Infocom'97*, Kobe, Japan, Apr. 1997.