A Queueing Model for Wormhole Routing with Timeout†

Po-chi Hu and Leonard Kleinrock
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095-1596

Abstract

In this paper, we propose an analytical model for wormhole routing with the use of a timeout reset mechanism. This model is based on an M/G/1 queueing system with impatient customers and feedback. Some approximations are proposed and verified by simulation. By comparing our analytical results to simulation, we show that the proposed model successfully captures the performance characteristics of wormhole routing with a timeout reset mechanism.

1 Introduction

Wormhole routing is a widely studied algorithm which has typically been applied to supercomputer interconnection networks. Recently wormhole routing has been used as the switching scheme for Local area networks (LANs). One such effort is Myricom's *Myrinet* [20], which has been adopted as the LAN infrastructure for the Supercomputer SuperNet (SSN), a research project being conducted at UCLA, JPL and Aerospace Corp. [14].

Many performance studies for wormhole routing in a supercomputer environment have been carried out and presented in the literature [1, 2, 5, 6, 9, 16]. However, many performance studies do not focus on the LAN environment, which suffers link propagation delay and has low-cost non-intelligent switches. Moreover, except for the simulation studies in [16], there is no analysis work evaluating the timeout reset mechanism, which is not only a powerful technique to solve potential deadlock problems but is also able to reduce blocking and achieve significant performance improvements for wormhole routing.

In this paper, an analytical model for wormhole routing with a timeout reset mechanism based on an M/G/1 queueing system with impatient customers and feedback is proposed and verified. An exact analysis of the link holding time distribution is developed and expressed in terms of Laplace-Stieltjes transforms. To derive solutions for these Laplace-Stieltjes transforms, an approximation method which matches the first two moments of the link waiting time (i.e., the blocking time) is proposed. To test this model, we performed some simulations to compare results as well as to verify the approximations used in the model. The only assumptions for those simulations are: exponential worm length, Poisson arrival, and that the bandwidth required for control signals (e.g. timeout reset signals) is negligible. The comparison results show that our model is general enough to capture a spectrum of networks using wormhole routing with timeout.

In section 2, we describe the basic network structure as well as wormhole routing. Then we develop the analysis model in sections 3–6. In section 7, the results and comparisons with simulations are presented, and a discussion of the accuracy of this model is given. Finally, section 8 contains the conclusion and future work.

2 Wormhole routing

In general, we consider a network for which all communication links are bi-directional and have the same capacity. Packets are generated and absorbed at hosts only. We assume that packet generation is a Poisson process and packet length is exponentially distributed. We measure packet length by flits, which is the amount of data that can be transmitted in one time unit. For example, the 640Mbps Myrinet has one byte per flit lasting 12.5ns. *Source routing* is employed because switches have no processing capability to maintain a routing table. A routing path, which specifies the links that a packet will traverse in order, is generated by the source host and attached to the head of the packet. Since switches have no intelligence (for low cost) and specifically can do no adaptive routing, hence routing paths don't change except at hosts when timeout retransmissions occur. An example network configuration is shown in figure 1. Also, as shown in

†This work was supported by the USDOD ARPA/CSTO under Contract DABT63-93-C-0055 The Distributed Supercomputer Supernet—A Multi Service Optical Intelligent Network.

0-8186-7180-7/95 $04.00 © 1995 IEEE

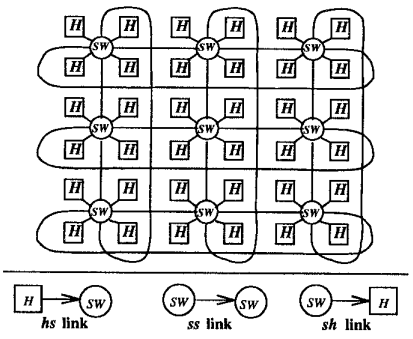

Figure 1: *An example network configuration:* 3×3 *torus.*

figure 1, we use "*hs* link", "*ss* link" and "*sh* link" to denote host-to-switch, switch-to-switch and switch-to-host links, respectively.

Wormhole routing is the basic switching technique we study. Wormhole routing was first introduced in [19]. It was developed from the earlier idea of *cut-through switching* [10]. In wormhole routing, switches have relatively small buffers. As opposed to store-and-forward switching, as soon as a packet header (or its routing information) is received, this packet is forwarded to the next switch (before it is completely received); if the outgoing link to the next switch is busy serving another packet, our packet gets blocked and resides in the switch until the outgoing link is available. In this case, called *blocking*, the switch must inform the previous up-stream switch to stop transmission (i.e., it exercises *back-pressure flow control*) due to the limited size of buffers. A packet (which is also called a *worm*) might be buffered in several nodes along the chain while stuck in the middle of the network due to blocking. With wormhole routing, deadlocks are possible unless a deadlock-free routing strategy is employed. A survey of wormhole routing can be found in [15].

Backward timeout reset is the basic mechanism we study to solve deadlock problems. Whenever a worm head reaches a switch, a timer starts counting how long this worm resides at this switch while waiting for its outgoing link to become available (thus advancing to the next switch or host node). If this "residence time" exceeds a timeout threshold, then a timeout event is triggered; a switch at which timeout occurs will then clear all buffers occupied by this worm and will issue a timeout reset signal backward to the upstream node from which this worm came. A switch which receives a timeout reset signal will pass this signal further upstream and will also free the outgoing link and any buffer occupied by this timed-out worm. This process continues until the timeout reset signal reaches the source host where the worm was generated. (We assume that a switch can always send the timeout reset signal upstream even if the tail of the worm has already left this switch). The source host, after receiving the timeout reset signal, will stop the transmission of this worm if the transmission is still in progress, and will insert the worm back into the tail of this host's packet queue so that it will be retransmitted later.

In this study, the routing strategy is not specified. It could be shortest-path routing or any deadlock free routing. Also, the network topology could be regular or arbitrary.

3 Traffic analysis

In this section, we systematically analyze the traffic rate at each transmission link. The analysis is simply based on the timeout probability at each link and the traffic arrival rates of all routing paths. For convenience of presentation, we use the notation in table 1 throughout the paper.

N_p	:	The total number of routing paths.
γ	:	The total external worm arrival rate.
δ_p	:	The probability that an external arrival worm is routed to path p.
h_p	:	The length (number of hops) of path p.
l_i	:	Link i.
l_{pi}	:	The ith link of path p; $1 \leq i \leq h_p$. If the ith link of path p is link k, then $l_{pi} \equiv l_k$.
λ_p	:	The total worm arrival rate on path p.
λ_{l_i}	:	The total worm arrival rate at link i.
λ_{pi}	:	The arrival rate at link i of worms via path p.
$P_{T_{l_i}}$	:	The probability of timeout on link l_i.
$P_{T_{l_{pi}}}$	:	The probability of timeout on link l_{pi}.
P_{F_p}	:	The transmission failure probability of path p; the probability that a worm on path p will be timed-out in its current transmission.
P_{S_p}	:	The transmission success probability of path p. $P_{S_p} = 1 - P_{F_p}$
$\mathcal{L}_p$	:	The set of links which are traversed along path p.

Table 1: *The notation for traffic analysis.*

We assume that there are N_p possible paths in the network, and that the arrival rate for path p is $\gamma\delta_p$. Here, γ is the total worm arrival rate to the network, which we call *the external worm arrival rate*, and δ_p

$\omega_{l_i}, \omega_{l_{pk}}$	:	Random variables which denote the waiting time at l_i and l_{pk} respectively.
ℓ	:	A random variable which denotes the worm length.
$W_{l_i}(x)$	:	The probability distribution function (PDF) of ω_{l_i}.
$W^*_{l_i}(s)$	:	The Laplace-Stieltjes transform of $W_{l_i}(x)$.
$B_{S_{pk}}(x)$	:	The link holding time distribution for worms traversing path p at their kth link, given worms successfully reach their destinations.
$B^*_{S_{pk}}(s)$	:	The Laplace-Stieltjes transform of $B_{S_{pk}}(x)$.
$B_{F_{pk\|j}}(x)$	:	The link holding time distribution for worms traversing path p at their kth link, given that these worms will suffer timeouts j hops from now.
$B^*_{F_{pk\|j}}(s)$	:	The Laplace-Stieltjes transform of $B_{F_{pk\|j}}(x)$.
$B^*_{pk}(s)$	:	The Laplace-Stieltjes transform of the link holding time distribution for worms traversing path p at their kth link.
b_{l_i}	:	A random variable which denotes the link holding time of link i.
$B_{l_i}(x)$	:	The link holding time distribution of link i.
$B^*_{l_i}(s)$	:	The Laplace-Stieltjes transform of $B_{l_i}(x)$.
$L^*(s)$	:	The Laplace-Stieltjes transform of the worm length distribution.
$P_{S_{pk}}$	:	The probability of transmission success for worms traversing path p, given that they are already on their kth link. $P_{S_{pk}} = \prod_{j=k+1}^{h_p}(1 - P_{T_{l_{pj}}})$.
$P_{F_{pk\|j}}$	:	The probability of transmission failure j hops from now for worms traversing path p and already on their kth link. $P_{F_{pk\|j}} = P_{T_{l_{pj+k}}} \prod_{i=k+1}^{k+j-1}(1 - P_{T_{l_{pi}}})$.

Table 2: *The notation for link holding time distributions.*

is the probability that the external arriving worm is routed to path p. Since a worm may possibly timeout during its transmission, in which case it will have to be retransmitted later, we say we have a *transmission failure* when a worm times-out, and similarly, a *transmission success* when a worm successfully reaches its destination.

If there is no timeout, no matter what the network topology or routing strategy is, we can derive the arrival rate at a particular link simply by summing up the arrival rates on paths which traverse this link. With the possibility of timeout, it still can be expressed as:

$$\lambda_{l_i} = \sum_{\forall p, l_i \in \mathcal{L}_p} \lambda_{pi} \tag{1}$$

Now, let $\xi_p(i)$ be a function which returns k if link i is the kth link of path p, or 0 if $l_i \notin \mathcal{L}_p$. With the assumption that the probability of timeout at each link is independent of the arriving worms, λ_{pi} can easily be derived as a product form of probabilities that a worm does not suffer timeout at links prior to link i:

$$\lambda_{pi} = \begin{cases} \lambda_p \prod_{j=1}^{\xi_p(i)-1}(1 - P_{T_{l_{pj}}}) & \text{if } l_i \in \mathcal{L}_p \\ 0 & \text{otherwise} \end{cases}$$

To derive the total arrival rate on path p, we first have to derive the transmission success probability P_{S_p}, and the failure probability P_{F_p}. Since P_{S_p} is simply the probability that no timeout occurs, we have,

$$P_{S_p} = \prod_{\forall k, l_k \in \mathcal{L}_p} (1 - P_{T_{l_k}}) \tag{2}$$

and

$$P_{F_p} = 1 - P_{S_p} = 1 - \prod_{\forall k, l_k \in \mathcal{L}_p} (1 - P_{T_{l_k}}) \tag{3}$$

By summing up the external arrival rate and the timeout retransmissions, we get

$$\begin{aligned} \lambda_p &= \gamma\delta_p + \gamma\delta_p P_{F_p} + \gamma\delta_p P_{F_p}^2 + \cdots \\ &= \frac{\gamma\delta_p}{P_{S_p}} \end{aligned}$$

Finally, substituting these results into equation (1), we get

$$\lambda_{l_i} = \sum_{\forall p, l_i \in \mathcal{L}_p} \frac{\gamma\delta_p \prod_{j=1}^{\xi_p(i)-1}(1 - P_{T_{l_{pj}}})}{\prod_{\forall k, l_k \in \mathcal{L}_p}(1 - P_{T_{l_k}})} \tag{4}$$

The remaining unknown variable is $P_{T_{l_{pj}}}$ and $P_{T_{l_k}}$ (the probability of timeout at each link), which we consider in section 4.

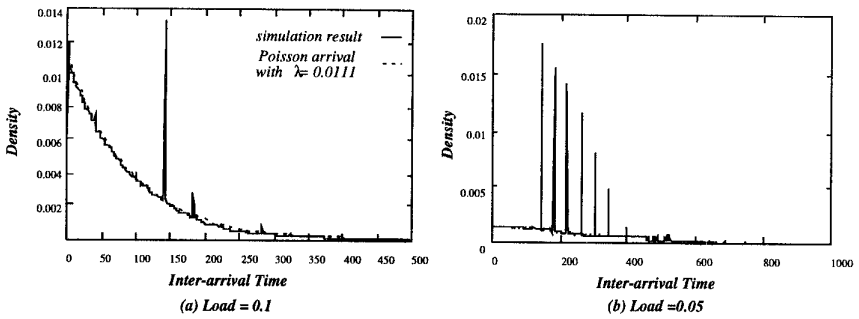

Figure 2: *The inter-arrival time distribution (by simulation) of a switch-to-switch outgoing link in a 7 × 7 torus network. (average worm length = 100 flits, timeout = 100 time units, propagation delay = 20 time units)*

4 Link Holding Time

Link holding time is the interval from when a worm first grabs a link until this worm releases it. The notation for link holding time and waiting time distributions are defined in table 2. Calculating the distribution of link holding time is difficult because of possible blocking; link holding time is not only a function of worm length but also depends upon the waiting time (blocking time) for all links in the path, and this depends on link holding time itself. Furthermore, the timeout mechanism makes it more complicated since the transmission may be aborted. Fortunately, the following observations allow us to find a good approximation to the distribution of link holding time:

- Simulation shows that the inter-arrival time distribution at each outgoing link is nearly exponential (e.g. a Poisson process) for a wormhole routing environment, especially under heavy load (see figure 2). At light loads, simulations show that some huge spikes caused by the timeout mechanism exist; except for these spikes, it is close to exponential. However, since the effect of blocking is not significant at light loads, we believe that the Poisson arrival approximation is reasonable[‡].
- One difficult problem in traditional store-and-forward networks is the dependence between the inter-arrival time and the service time (link holding time) at all links (except the host-to-switch links). To solve this difficulty, we will use the Kleinrock's *independence assumption* [11], which allows us to choose a new service time for each packet received. This assumption works well in our case because of the cut-through feature of wormhole routing; namely, a worm can be forwarded to the next node before it is completely received. Therefore, a long (short) inter-arrival time does not imply a long (short) link holding time for the arriving worm in wormhole routing.
- With Poisson arrivals and Kleinrock's *independence assumption*, the process on a single link is simply an *M/G/1 queueing system with impatient customers* (also referred to as *M/G/1 with reneging*), a system that has been well studied (for example, see [3, 4, 18]). Some elegant solutions to these systems are available.
- From simulation, we also found that the profile of a *link waiting time* distribution is surprisingly simple (figure 3). This suggests the possibility of using some simple rational functions to approximate the actual waiting time distribution. As shown in figure 3, in some cases the density of waiting time is almost a straight line.

Since a served worm releases a link only when it sends out its tail at this link or receives the timeout backward reset signal, there are two cases for link holding time.

Successful case: If a worm successfully reaches its destination, its link holding time is the time spent in blocking (waiting) for the rest of its trip plus the worm length (see figure 4). Therefore, we have the distribution function,

$$B_{S_{pk}}(x) = \mathbf{Prob}\left\{\omega_{l_{pk+1}} + \omega_{l_{pk+2}} + \cdots + \omega_{l_{ph_p}} + \ell \le x\right\}$$

and, the Laplace-Stieltjes transform:

$$B^*_{S_{pk}}(s) = L^*(s) \prod_{j=k+1}^{h_p} W^*_{l_{pj}}(s) \qquad (5)$$

[‡]Actually, the arrival is not a Poisson process, even though the inter-arrival time is exponentially distributed. This is discussed in section 7.

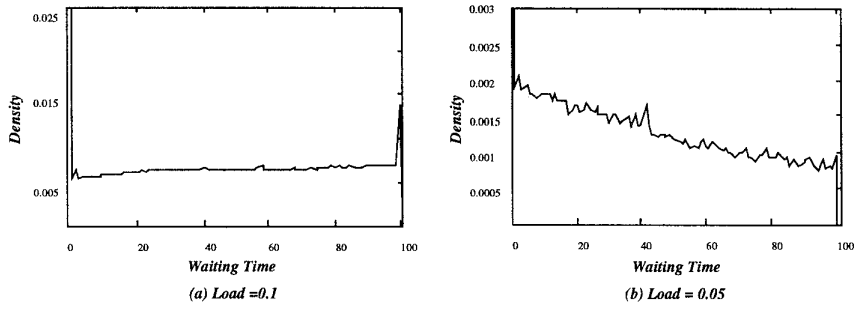

Figure 3: *The waiting time distribution (by simulation) for an ss outgoing link in a 7 × 7 torus network. (average worm length = 100 flits, timeout = 100 time units)*

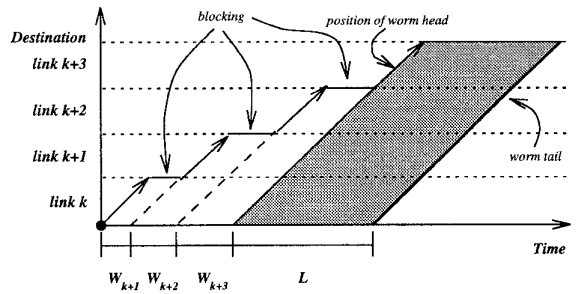

Figure 4: *an illustration of the link holding time for a successful transmission.*

Timed-out case: When a worm suffers transmission failure, it will hold a link until it times-out at this link, or until its tail is sent out of this link before it receives the timeout signal. We assume that worms are long enough so that we can ignore the situation where the tail of a worm leaves a link before it receives the timeout reset signal (i.e., the second case in the previous sentence). Then, we have that the link holding time distribution, conditioned on the link where timeout occurs, is the following:

$$B_{F_{pk|j}}(x) = \mathbf{Prob}\{\tau_{l_{pk}} + \omega_{l_{pk+1}} + \tau_{l_{pk+1}} + \omega_{l_{pk+2}} + \cdots + \omega_{l_{pk+j-1}} + \tau_{l_{pk+j-1}} + \tau_t + \tau_{l_{pk+j-1}} + \tau_{l_{pk+j-2}} + \cdots + \tau_{l_{pk}} \le x\}$$

where τ_t denotes the timeout interval and $\tau_{l_{pk}}$ represents the propagation delay of the kth link of path p.

The above equation is directly derived by observing figure 5. Consider a worm that will fail. After it grabs a link, it forwards to the next node along with a propagation delay. Then it waits for the next link to become available so it can forward to the next node, etc., until it reaches the node where its timeout occurs. At this timeout node, the worm head dwells there for the timeout period, and then releases its holding links by sending back the timeout reset signal, which again, will suffer a series of propagation delays.

Assuming that both timeout and the propagation delay are deterministic, we have the Laplace-Stieltjes transform of $B_{F_{pk|j}}(x)$ directly from above results:

$$B^*_{F_{pk|j}}(s) = e^{-s(\tau_t + 2\tau_{l_{pk}})} \prod_{i=k+1}^{k+j-1} W^*_{l_{pi}}(s) e^{-2s\tau_{l_{pi}}} \quad (6)$$

From equations (5) and (6), and by unconditioning on success or failure at j hops from now, we have,

$$\begin{aligned} B^*_{pk}(s) &= P_{S_{pk}} B^*_{S_{pk}}(s) + \sum_{j=1}^{h_p-k} P_{F_{pk|j}} B^*_{F_{pk|j}}(s) \\ &= L^*(s) \prod_{j=k+1}^{h_p} \left[(1 - P_{T_{l_{pj}}}) W^*_{l_{pj}}(s)\right] \\ &\quad + \sum_{j=1}^{h_p-k} \left(\left[P_{T_{l_{pj+k}}} e^{-s(\tau_t + 2\tau_{l_{pk}})}\right] \times \prod_{i=k+1}^{k+j-1} \left[(1 - P_{T_{l_{pi}}}) W^*_{l_{pi}}(s) e^{-s\tau_{l_{pi}}}\right] \right) \quad (7) \end{aligned}$$

Then, finally a link holding time distribution is given through

$$B^*_{l_i}(s) = \sum_{\forall p, l_i \in \mathcal{L}_p} \frac{\lambda_{pi}}{\lambda_{l_i}} B^*_{p\xi_p(i)}(s) \quad (8)$$

where $\frac{\lambda_{pi}}{\lambda_{l_i}}$ simply gives the probability that an arriving worm at link i traverses path p.

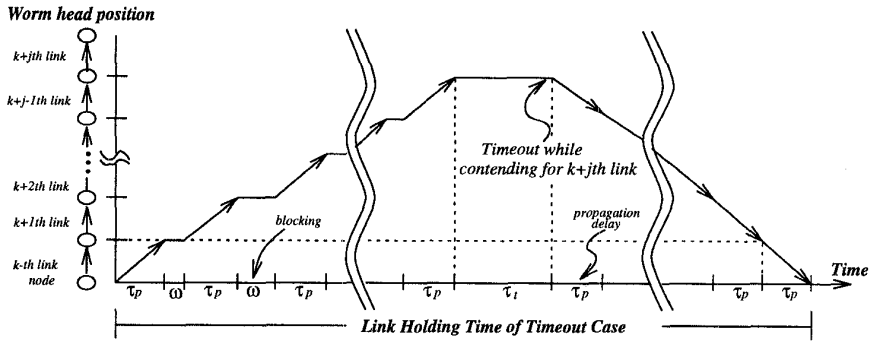

Figure 5: *The forwarding process of a worm in a timeout case.*

5 From holding time to waiting time

In section 4, we derived the link holding time distribution $B^*_{l_i}(s)$, if $\forall j, W^*_{l_j}(s)$ is known. Now, the remaining problem is to find link waiting time distributions.

As discussed at the beginning of section 4, the queueing process at each outgoing link is close to an M/G/1 queueing system with impatient customers. Using the results for an M/G/1 queueing system with deterministic reneging time developed in [4], the waiting time distribution at link i, given no timeout, is

$$W_{l_i}(x) = \begin{cases} \frac{\phi_i(x)}{\phi_i(\tau_t)} & 0 \le x \le \tau_t \\ 1 & x > \tau_t \end{cases} \quad (9)$$

where $\phi_i(x)$ is a function which satisfies

$$\Phi^*_i(s) \triangleq \int_0^\infty e^{-sx} d\phi_i(x) = \frac{s\phi_i(0)}{s - \lambda_{l_i} + \lambda_{l_i} B^*_{l_i}(s)} \quad (10)$$

and

$$\phi_i(0) + \lambda_{l_i} \overline{b_{l_i}} \phi_i(\tau_t) = 1 \quad (11)$$

where $\overline{b_{l_i}}$ is the mean of b_{l_i} (the link holding time of link i). Note that $\phi_i(x)$ is not a distribution function.

Unfortunately, it is difficult to find $B^*_{l_i}(s)$ and $W^*_{l_i}(s)$ directly from equation (8) and the above equations. Therefore, the following approximation method is suggested.

Two moment matching approximation: The basic idea is that any distribution function can be approximated arbitrarily closely by a *series-parallel stage-type* device [13]. As discussed at the beginning of section 4, it is reasonable to approximate a link waiting time distribution by only a few exponential stages since it has a very simple profile. Here, we choose to use a single-stage approximation in which α_i and β_i must be selected to match the first two moments of $W^*_{l_i}(s)$. Thus, we assume:

$$\frac{d\phi_i(x)}{dx} = \alpha_i e^{-\beta_i x} \quad \text{for } x > 0 \quad (12)$$

$$\phi_i(x) = \frac{\alpha_i}{\beta_i}(1 - e^{-\beta_i x}) + \phi_i(0) \quad (13)$$

Note that β_i may have a negative value since the utilization factor, $\rho_{l_i} = \lambda_{l_i} \overline{b_{l_i}}$, can be greater than one in deterministic reneging queueing systems [4].

Certainly, we may choose more stages to match higher moments for a more precise approximation. However, the approximation of a Poisson arrival process plus the fact that the profile of link waiting time distributions is simple, suggests that a more sophisticated approximation with more stages is probably not justified.

From equation (9), we may calculate the (approximated) first moment of the waiting time distribution $\overline{w_{l_i}}$, as

$$\begin{aligned} \overline{w_{l_i}} &= \int_0^{\tau_t} x dW_{l_i}(x) \\ &= \int_{0+}^{\tau_t} \frac{x\alpha_i e^{-\beta_i x}}{\phi_i(\tau_t)} dx \\ &= \frac{\alpha_i(1 - e^{-\beta_i \tau_t} - \beta_i \tau_t e^{-\beta_i \tau_t})}{\beta_i^2 \phi_i(\tau_t)} \end{aligned} \quad (14)$$

and similarly, we can derive the second moment of the approximated $W_{l_i}(x)$.

Now, $\phi_i(0)$ and $\phi_i(\tau_t)$ are simply found from equations (11) and (13), to give us

$$\phi_i(\tau_t) = 1 - \frac{\lambda_{l_i} \overline{b_{l_i}} - \frac{\alpha_i}{\beta_i}(1 - e^{-\beta_i \tau_t})}{1 - \lambda_{l_i} \overline{b_{l_i}}}$$

$$\phi_i(0) = 1 - \lambda_{l_i}\overline{b_{l_i}} + \frac{\lambda_{l_i}\overline{b_{l_i}}\left[\lambda_{l_i}\overline{b_{l_i}} - \frac{\alpha_i}{\beta_i}\left(1 - e^{-\beta_i\tau_t}\right)\right]}{1 - \lambda_{l_i}\overline{b_{l_i}}}$$

Actually, $\phi_i(\tau_t)$ is exactly the probability that an arriving worm will not suffer a timeout, and $\phi_i(0)$ is the probability that an arriving worm finds link i idle (no blocking) [4]. Hence the timeout probability is given by

$$P_{T_{l_i}} = 1 - \phi_i(\tau_t) = \frac{\lambda_{l_i}\overline{b_{l_i}} - \frac{\alpha_i}{\beta_i}\left(1 - e^{-\beta_i\tau_t}\right)}{1 - \lambda_{l_i}\overline{b_{l_i}}} \quad (15)$$

Remember that $\phi_i(x)$ is not a distribution function; $\int_0^\infty d\phi_i(x)$ is not necessarily equal to one. Therefore, two moment matching is required to solve for α_i and β_i. From equation (8), by differentiating and setting $s = 0$, we find the moments of $B_{l_i}(x)\,\forall i$, in terms of the moments of $W_{l_j}(x)\,\forall j$. Using the series-parallel stage approximation of $W_{l_i}(x)\,\forall i$, we can compute the moments of the approximated $W_{l_i}(x)\,\forall i$ (e.g. equation (14)). Then, the moments of $\phi_i(x)\,\forall i$ can be found directly from equation (10). By forcing the moments of $\phi_i(x)\forall i$ found from above to be equal to the ones directly derived from equation (13) (by definition of the moments), we finally get a set (unfortunately, a huge set) of equations in terms of α_i and β_i, $\forall i$. Solving these equations together with equation (15) (the probability of timeout) and equation (4) (the result of traffic analysis), we obtain α_i and β_i, and then $P_{T_{l_i}}$, λ_{l_i}, $B_{l_i}(x)$ as well as $W_{l_i}(x)\forall i$. As usual, extensive numerical calculations are required to iteratively find the feasible roots that must be evaluated in this process [13].

6 Host queueing time

The queueing process at a host is basically an M/G/1 queueing system with feedback (see figure 6). The exact solution for the waiting time distribution of this queueing system can be found in [7, 8, 17]. However, the exact solution is too complicated to be practical. Fortunately, what we are interested is only the mean waiting time, and it can be derived simply by changing the order of service (shown in figure 7) and noting from the conservation law [12] that the mean wait is the same for figures 6 and 7. Thus, we simplify the host queueing process to a pure M/G/1 model.

The probability of feedback is the probability that a transmission fails because of a timeout. Thus,

$$P_{S,q_a} = \frac{\sum_{\forall p, p\in\mathcal{P}_a}\lambda_p P_{S_p}}{\sum_{\forall p, p\in\mathcal{P}_a}\lambda_p} \quad (16)$$

$$P_{F,q_a} = 1 - P_{S,q_a} \quad (17)$$

where $\mathcal{P}_a$ is the set of paths that start at host a, and P_{S,q_a}, P_{F,q_a} are the probabilities of transmission

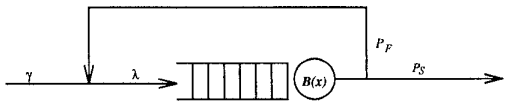

Figure 6: *The queueing process at a host.*

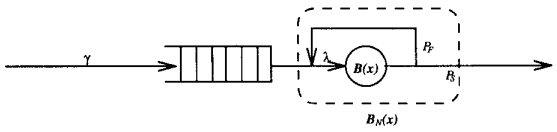

Figure 7: *The simplified host queue model.*

success and failure respectively, for all transmissions started at host a.

Let $b_{q_a,n}$ be a random variable representing the service time of the nth try of a transmission for worms from host a. Also, $B_{q_a}(x)$, $B^*_{q_a}(s)$ and $B_{N_{q_a}}(x)$, $B^*_{N_{q_a}}(s)$ are the corresponding distribution functions and the Laplace-Stieltjes transforms of the service time for one try of a transmission in the original M/G/1 system, and the total service time in the simplified pure M/G/1 queueing model respectively. It is clear that in the original M/G/1 system, the service time of one transmission try is exactly the link holding time of the connected host-to-switch link at this host. Therefore, we have the service time distribution of one try directly:

$$B^*_{q_a}(s) = B^*_{l_i}(s)$$

if link i is exactly the hs link that connects to host a.

For the simplified pure M/G/1 model, we have

$$B_{N_{q_a}}(x) = \sum_{n=1}^{\infty} P_{S,q_a}P^{n-1}_{F,q_a}\mathbf{Prob}\left\{b_{q_a,1} + b_{q_a,2} + \cdots + b_{q_a,n} \le x\right\}$$

and

$$B^*_{N_{q_a}}(s) = \sum_{n=1}^{\infty} P_{S,q_a}P^{n-1}_{F,q_a}B^*_{q_a}(s)^n = \frac{P_{S,q_a}B^*_{q_a}(s)}{1 - P_{F,q_a}B^*_{q_a}(s)} \quad (18)$$

Now, by applying the *Pollaczek-Khinchin (P-K) mean-value formula* [13], we get the mean total waiting time,

$$\overline{w_{N_{q_a}}} = \frac{\gamma_{q_a}\overline{b^2_{N_{q_a}}}}{2(1 - \rho_{N_{q_a}})} \quad (19)$$

where γ_{q_a} is the total external arrival rate at host a, and $\gamma_{q_a} = \sum_{\forall p, p \in \mathcal{P}_a} \gamma \delta_p$. The utilization factor, $\rho_{N_{q_a}} = \gamma_{q_a} \overline{b_{N_{q_a}}}$, and $\overline{b_{N_{q_a}}}$, $\overline{b^2_{N_{q_a}}}$ are the first and second moments of $B_{N_{q_a}}(x)$, which can be found directly by differentiating $B^*_{N_{q_a}}(s)$ and setting $s = 0$. Note that we implicitly make an important assumption that each try of a transmission has an independent service time and independent failure or success, which is generally not true. However to closely model the dependence of service time between successive tries is far from trivial. In section 7, we discuss more about this assumption.

At this point, we have developed the models and approximations to derive $\overline{b_{N_{q_a}}}$, the mean total service time for a transmission, and $\overline{w_{N_{q_a}}}$, the mean total waiting time in the host queue. The average network latency (the mean time from when worm is generated at the source host to the instant when the destination node receives the whole worm) for path p traffic starting at host a, $\overline{T_{qa}}$, is simply derived as:

$$\overline{T_{qa}} = \overline{w_{N_{q_a}}} + \overline{b_{N_{q_a}}} + \sum_{\forall j, l_j \in \mathcal{L}_p} \tau_{l_j} \qquad (20)$$

where $\sum_{\forall j, l_j \in \mathcal{L}_p} \tau_{l_j}$ simply counts the propagation delay for the worm's tail, after it leaves the source host, to reach the destinated node.

7 Verification

In this section, we show results from the analytical model and compare them to those obtained from simulation. The comparison is for the 3×3 torus topology, as shown in figure 1. Results with different timeout values and worm lengths are shown in figures 8 and 9.

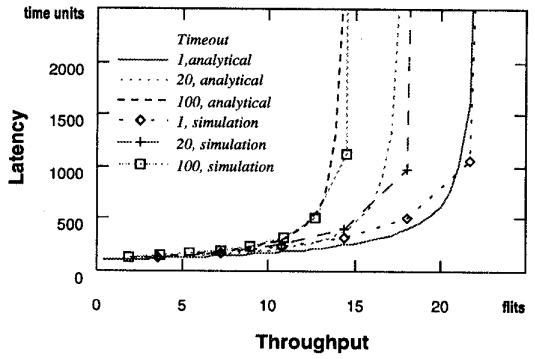

Figure 8: *Results of the analytical model with different timeouts. (worm length = 100 flits, propagation = 1 time unit)*

In figure 8, we find that the analytical results are close to those from simulation, even at very high loads.

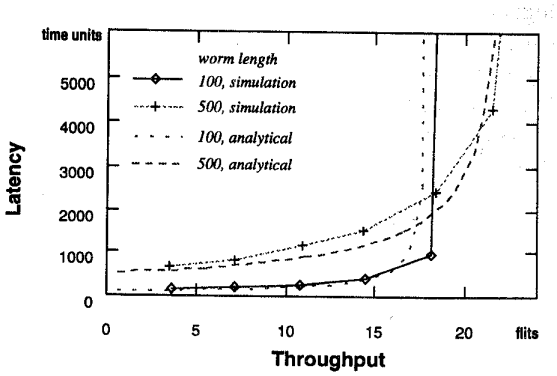

Figure 9: *Results of the analytical model with different worm lengths. (timeout = 20 time units, propagation = 1 time unit)*

Both the analytical and the simulation results point out that a low timeout gives higher throughput.

Figure 9 shows the results for different worm lengths. Again, results from the analytical model and simulation are close over the whole performance spectrum. The analytical model successfully points out that longer worms result in higher throughput but generate longer delays at low traffic loads. The maximum throughput derived from the analytical model in both figures 8 and 9 are within 10 percent of the simulation. The delays are almost identical for both simulation and analysis in light to moderate load regions; this is the case when the worm length is short and the timeout threshold is not too small.

If we look at figures 8 and 9 more closely, we find that the analytical model always underestimates both the average delay in the medium load region and the maximum network throughput, as compared to the simulation results. This difference is mainly due to two assumptions in the model: the independence assumption between successive retransmissions for the host queueing time analysis, and the Poisson arrival process for each output link.

As mentioned in section 6, we made an important assumption regarding the host queueing time analysis. Namely, we assumed that the result of a worm's timeout retransmission is independent of the worm's previous transmission attempt. Certainly, this assumption is false, especially when the worm length is long and the timeout is small. In fact, for each value of timeout, a worm, if it is retransmitted immediately, is very likely to get blocked again at the same place where it timed-out in the previous try. Therefore, the probability of getting timed-out or suffering blocking is much

higher than for the assumed independent case. This clearly results in an underestimate of delay in our analytical model. Because a long worm implies a longer link holding time and a small timeout makes the retransmission instants closer to each other, the dependency is stronger in the above cases. This explains why the delay is underestimated in figures 8 and 9, when the worm is long or the timeout is very short.

However, the dependency between successive retransmission does not affect the maximum throughput. Since the host queues build up quickly when the offered load is close to the saturation point, the retransmission of a timed-out worm occurs far later than the timeout instant due to the huge waiting time in the host queue. Hence, the dependence of these two instants is broken, and the maximum throughput is not changed by this dependence of successive retransmissions.

In figure 10, we see the results when we add a constant delay for all retransmissions. This constant delay simply relaxes the dependence between the timeout and retransmission instants, and therefore, as we can see from the figure, the difference between analytical and simulation results is diminished in the delayed retransmission case. Nevertheless, the maximum throughput is not changed.

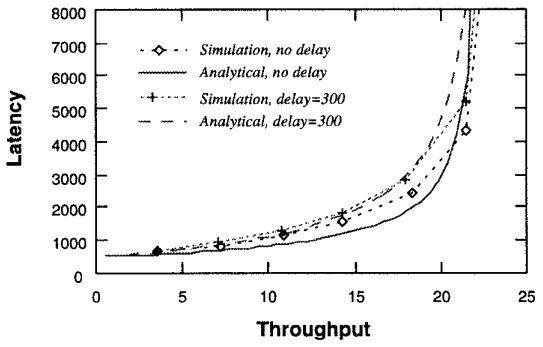

Figure 10: *Results of delayed retransmission. (worm length = 500, timeout = 20, propagation = 1)*

The reasons for the underestimated maximum throughput are mainly from the following:

- **The buffer effect** is not modeled in the analytical model. In the actual switches and in the simulation, there is a minimum buffer which stores the data currently being propagated across the link. This buffer absorbs part of the worm and consequently reduces the blocking, especially, when the propagation delay is non-negligible. As shown in figure 11, with a larger buffer and a longer propagation delay, there is a great difference between the analytical model and the simulation result.

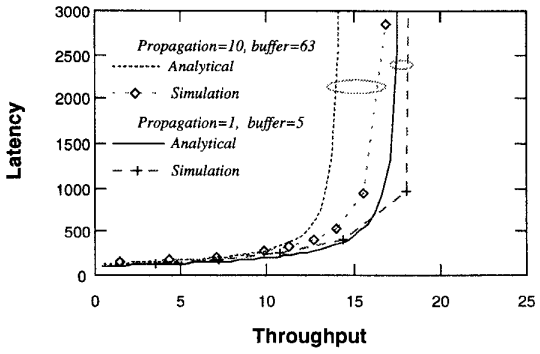

Figure 11: *The buffer effect on network performance. (worm length = 100, timeout = 20)*

- A major error is introduced by **the assumption of a Poisson arrival process** at each link. From the simulation, as shown in section 4, we already found that the inter-arrival time of worms is close to an exponential distribution. However, the arrival process is dependent on the link status. It is clear that for an 8×8 switch, there can be at most 7 worms requesting the same link (assuming that no worm will try to leave on the same link as which it entered a switch). If we simply use exponential distribution for the link holding time, then the queueing process of each link is Markovian as shown in figure 12. The actual worm arrival rate decreases as the number of worms waiting for this same link increases. It can be shown that the above Markov process has the same inter-arrival time distribution as the Poisson assumption; however, it results in less average waiting time before getting service (figure 13). Thus, the link holding time and hence the timeout probability is overestimated in our analytical model. Consequently, the maximum throughput is underestimated.

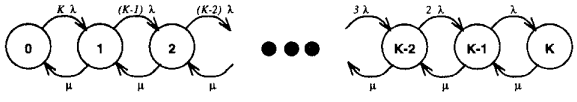

Figure 12: *An illustration of the actual queueing process on an output link.*

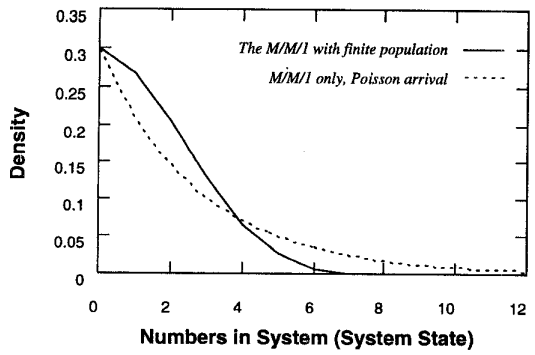

Figure 13: *The comparison of the state probabilities for a pure M/M/1 and an M/M/1 with finite population. (utilization = 0.7, 8 × 8 switch)*

8 Conclusion and Future Work

In this paper, we have developed a sophisticated analytical model, for wormhole routing with timeout reset. This model captures the whole spectrum of network performance as demonstrated by comparison to simulation results, and is general for any kind of network configuration. However, it is fairly complicated; extensive numerical calculations are required to iteratively find the solutions.

Many improvements and modifications are currently being considered for this analytical model. First, we are working on extending this model to include the effect of buffers, so it can closely describe the behavior of networks with non-negligible input buffer sizes. Second, the modifications of this model to account for the arrival process at links and the dependence between successive retransmissions are planned. Finally, we are seeking a simpler model more suitable for practical applications.

References

[1] V. S. Adve and M. K. Vernon, "Performance Analysis of Mesh Interconnection Networks with Deterministic Routing", *IEEE Trans. on Parallel and Distributed Systems,* vol. 5, no. 3, March 1994.

[2] A. Agarwal, "Limits on Interconnection Network Performance", *IEEE Trans. on Parallel and Distributed Systems,* vol. 2, no. 4, Oct. 1991.

[3] F. Baccelli, P. Boyer and G. Hebuterne, "Single-Server Queues with Impatient Customers", *Advance Applied Probability* **16**, pp.887–905, 1984.

[4] D. J. Daley, "General Customer Impatience in the Queue GI/G/1", *Journal of Applied Probability* **2**, pp.186–205, 1965.

[5] W. J. Dally, "Performance Analysis of K-ary n-cube Interconnection Networks", *IEEE Trans. on Computers,* vol. 39, no. 6, June 1990.

[6] J. T. Draper and J. Ghosh, "A simple Analytical Model for Wormhole Routing in Multicomputer Systems", *Journal of Parallel and Distributed Computing,* vol. 23, pp. 202–214, Nov. 1994.

[7] D. H. J. Epema, "Mean Waiting Times in a General Feedback Queue with Priorities", *Performance Evaluation* **13**, pp.45–58, Sep. 1991.

[8] B. T. Doshi, J. S. Kaufman, "Sojourn Time in an M/G/1 Queue", *Queueing Theory and its Applications,* North-Holland, 1988. pp. 207–233.

[9] J. Kim and C. R. Das, "Hypercube Communication Delay with Wormhole Routing", *IEEE Trans. on Computers,* vol. 43, no. 7, July 1994.

[10] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique", *Computer Networks,* vol. 3, pp.267–289, 1979.

[11] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay,* MgGraw-Hill (New York), 1964. Reprinted by Dover Publications, 1972.

[12] L. Kleinrock, "A Conservation Law for a Wide Class of Queueing Disciplines", *Naval Research Logistics Quarterly,* vol. 12, no. 2, pp.181–192, June 1965.

[13] L. Kleinrock, *Queueing Systems, Vol. I: Theory,* Wiley Interscience (New York), 1975.

[14] L. Kleinrock, et. al, "OPTIMIC: a scalable distributed all-optical terabit network", to appear in *Journal of High Speed Networks: special issue on Optical Networks,* 1995.

[15] L. M. Ni and P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", *Computer,* pp.62–76, Feb. 1993.

[16] D. S. Reeves and E. F. Gehringer, "Adaptive Routing for Hypercube Multiprocessors: A Performance Study", *International Journal of High Speed Computing* pp.1-29, March 1994.

[17] K. Rege, "On the M/G/1 Queue with Bernoulli Feedback", *Operations Research Letters* **14**, pp.163–170, Oct. 1993.

[18] R. E. Stanford, "Reneging Phenomena in Single Channel Queues", *Mathematics of Operations Research,* vol. 4, no. 2, May 1979.

[19] C. Seitz *et al.*, "The hypercube communications chip", *Dep. Computer Science, California Inst. Technol.,* Display File 5128:DF:85, March 1985.

[20] C. Seitz, D. Cohen and R. Felderman, "Myrinet—A Gigabit-per-second Local-Area Network", *IEEE Micro,* vol. 15, no. 1, pp. 29–36, Feb. 1995.