Limits on Interconnection Network Performance

Anant Agarwal
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

As the performance of interconnection networks becomes increasingly limited by physical constraints in high-speed multiprocessor systems, the parameters of high-performance network design must be reevaluated, starting with a close examination of assumptions and requirements. This paper models network latency, taking both switch and wire delays into account. A simple closed form expression for contention in buffered, direct networks is derived and is found to agree closely with simulations. The model includes the effects of packet size and communication locality. Network analysis under various constraints (such as fixed bisection width, fixed channel width, and fixed node size) and under different workload parameters (such as packet size, degree of communication locality, and network request rate) reveals that performance is highly sensitive to these constraints and workloads. A two-dimensional network has the lowest latency only when switch delays and network contention are ignored, but three or four dimensions are favored otherwise. However, two-dimensional networks regain their advantage if communication locality exists. Communication locality decreases both the base network latency and the network bandwidth requirements of applications. We show that a much larger fraction of the resulting performance improvement arises from the reduction in bandwidth requirements than from the decrease in latency.

1 Introduction

An efficient communication network for high-performance multiprocessors must provide low latency memory access and message transmission. While some communication latency can be tolerated by overlapping computation with communication, latency imposes fundamental limits on the effectiveness of multiprocessors. Communication latency depends not only on the properties of the network, such as dimension, channel width, node delay, and wire delay, but on the communication patterns of parallel computations as well. This paper analyses the contribution of these factors to the latency of direct networks.

In a direct network [24], the processing nodes communicate directly with each other over a set of point-to-point links. The point-to-point interconnections between processors distinguish direct networks from indirect networks (or multistage networks) [27], such as the Omega [19] and the Delta [22] networks. An indirect network does not integrate processors and switches. Consequently, processors cannot communicate directly with each other, but must do so through a set of intervening switching nodes. Because they allow the exploitation of communication locality, direct networks are becoming increasingly popular for interconnections in large-scale concurrent computers. Examples of machines that use direct networks include the Caltech Cosmic Cube [25] and the Connection Machine [13].

We will focus on the general class of direct networks called k-ary n-cubes [28]. A k-ary n-cube is a network with n dimensions having k nodes in each dimension. For example, a 100 processor array has $n = 2$ and $k = 10$. Given N processors, the relationship $N = k^n$ holds between the dimension n and the radix k. For planar mappings, two or three-dimensional networks are favored because they scale better than high-dimensional networks, they are modular, and they are easy to implement. Examples of machine designs that use such networks are the MuNet [12], Ametek 2010 [26], the Caltech Mosaic [3], the MIT J-machine [9], and the CMU-Intel iWarp [4]. Some recent distributed shared-memory designs are also planning to use low-dimensional direct networks, e.g., HORIZON [18], the Stanford DASH Multiprocessor [20], and the MIT Alewife machine [2, 6].

The choice of the optimal network for a multiprocessor is highly sensitive to the assumptions about system parameters and the constraints that apply on the design. System parameters include, among other factors, message size and the degree of communication locality; design constraints include limits on bisection width, node size, and channel width. Bisection width is defined as the minimum number of wires that must be cut to separate the network into two equal halves [29]. A bisection width constraint is tantamount to an area constraint. A constraint on the node size is assumed to limit the number of pins on the node. Assuming a constraint on the bisection width, Dally [7, 8] analyzed the performance of k-ary n-cube networks implemented in two-dimensional space, using constant, logarithmic, and linear wire delay models. The analysis suggests that a two-dimensional network yields the lowest latency with a linear wire delay model. Node delays, however, were ignored (although the constant wire delay model does indicate the results when node delays are dominant), and message lengths used in obtaining the results were 150 and 200 bits, which are large for shared-memory machines, but typical for message passing multicomputers.

Node delays cannot be ignored with current technology. When node delays (or switch delays) are neglected and a constant bisection width is assumed, a network of lower dimension has lower latency for two reasons: (1) it has wider channels, resulting in smaller message sizes, and (2) it has shorter wires, resulting in faster packet transfer between switches. However, a lower-dimensional network forces a message to traverse more nodes. Because the effective network latency depends on both the node delay and the wire delay, longer wire delays might not be harmful if the node delays dominate. With current technology, this is indeed the case. For example, assuming an aggressive 10 nanosecond switch delay, it takes a wire length of about 10 feet for the wire delay to equal switch delay. Our results suggest that although two-dimensional networks have the lowest latency ignoring switch delay, three-dimensional networks are superior when switch delays are four times the wire delay of a two-dimensional network, and four-dimensional networks are best when the corresponding switch delays are 16 times greater.

Smaller messages diminish the relative advantage of networks with lower dimensions. When message sizes are small, wider channels – an advantage of low-dimensional networks – are less useful. Messages are expected to be smaller in a shared-memory multiprocessor (about 100 bits on average [5]) than in a message passing multicomputer. In addition, as observed in Section 3.3, small messages suffer less contention delay than large messages per unit volume of data transferred. Our analysis shows that small messages favor three-dimensional networks for large networks (greater than 1K nodes).

If bisection width is fixed to be the same as that of a unit-channel-width binary n-cube network with a thousand nodes, a two-dimensional network is clearly superior (see Figure 11(c)). However, if the node size is constrained to that of a two-dimensional network with 32 bit channels, a three dimensional network is optimal (see Figure 11(d)), while a four-dimensional network is slightly better than others when wire delays are the only constraining factor (see Figure 11(b)). Furthermore, in the previous two cases, higher dimensions are favored on account of their greater bandwidth, as the load on the network increases.

We show that communication locality in the application significantly improves both throughput and latency of direct networks, with a relative impact that increases with network load. We say a program running on a parallel machine displays *communication locality* (or memory reference locality) if the probability of communication (or access) to various nodes decreases with physical distance. Communication locality in parallel programs depends on the algorithms used as well as on the partitioning and placement of data and processes. When communication locality exists, low-dimensional networks outperform networks with higher dimensions. We compare the performance of direct and indirect networks under a node size constraint and show that low-dimensional direct networks do better than indirect networks only when communication locality exists.

Our analysis will examine network design based on several possible constraints. Technological constraints limit the wire density; we therefore assume a fixed wire density. Similarly, fundamental physical limitations on signal propagation speeds on wires will be maintained. Constant bisection width, however, is a potential limit imposed by cost, power, size, and other factors. Because bisection width is not a fundamental physical limit, this assumption will not always be made in this analysis. Instead, we will analyze networks under the following constraints:

- Constant channel widths
- Constant bisection width
- Constant node size

We develop a model for buffered low-dimensional direct networks that yields a simple closed form expression for network contention. (See [1] for a model of binary n-cube networks for unit packet sizes.) The model is thoroughly validated through measurements taken from a simulator. Although the assumptions made by the model are tailored to networks with high radices, in practice we have found that the model is accurate for low radix (e.g., $k = 4$) networks as well. Simple extensions to the model include the effects of packet size, multicycle pipelined switches, and communication locality.

We begin by presenting expressions for the base network latency that represent the effects of switch and wire delays in Section 2. The *base network latency* is the latency of an unloaded network. Section 3 derives a model for contention in buffered k-ary n-cube networks and validates it through simulations. Section 4 analyzes the effect of fixed channel widths, fixed bisection width, and fixed node size on the base network latency. Section 5 extends the analyses to include network contention and communication locality, and Section 6 summarizes the chief results of the paper.

2 Node Delays Versus Wire Delays

As advances in technology make the fabrication of ever-faster switches feasible, while wire speeds remain roughly constant, it is inevitable that wire delays will dominate switch delays. However, switches and wires of similar dimensions will have comparable delays because the same physical limits that govern wire speeds will also limit switch speeds. Therefore, our analysis includes the effect of both switch and wire delays.

The following argument describes the tradeoff that must be made in choosing a network topology. Assume that the clock cycle is chosen to be the sum of the switch delay and the wire delay in a synchronous system, making each network hop cost a cycle. (Other models could also be chosen to account for multicycle pipelined switches, or pipelined transmission channels.) The latency is a product of the clock cycle and the number of hops a message traverses. A higher dimensional network mapped onto a plane has longer wires, causing longer transmission times over each wire, but results in fewer hops.

2.1 Notation and Assumptions

Let message length in bits be L, network dimension be n, the radix be k, channel width in bits be W, and the number of nodes be N. Then, $N = k^n$, and message length in flits is L/W. In this paper, a flit is equal to the number of bits of data transferred over a channel in a clock cycle. Let T_b denote the base network latency (when contention is not considered), and let T_c denote the latency taking contention into account.

Message destinations are randomly chosen from all the nodes in the network unless specified otherwise. Although many network evaluation studies make this simplifying assumption, it is rarely true in practice. However, several software practices, such as memory interleaving, uniform distribution of parallel data structures, and distributed software combining tree implementations of barrier synchronizations, tend to spread accesses uniformly over all nodes. On the other hand, when software techniques are employed to enhance communication locality, non-uniform access patterns result; such inhomogeneous access distributions are considered in Section 5.1.

Let wire delay be denoted $T^w(n)$ and switch delay T^s. We assume that switch delay is a constant over various network topologies. This assumption is largely true for low-dimensional networks where a relatively large fraction of switch delay arises from factors such as chip crossings. For high-dimensional networks, the increased logic complexity will make switch delays sensitive to n, which makes the case for low-dimensional networks even more compelling. Of course, a more detailed analysis might assume that T^s is some function of the switch dimension. We will also assume a linear wire delay model. The switches are pipelined (i.e., switches use wormhole routing [7], which is a variant of cut through routing [14]). As mentioned before, the clock cycle is the sum of the switch delay and the delay due to the longest wire in the synchronous network.

Our study assumes that the networks are embedded in a plane. A similar analysis for mapping in three-dimensional space can be also be carried out, and we will suggest the changes needed when appropriate. We will consider networks with unidirectional channels and end-around connections. We will suggest the modifications necessary in the analyses to account for other topologies, such as networks with bidirectional channels, with and without end around connections. We have analyzed these alternate cases and we shall indicate instances where the results differ substantially from the network topologies considered in this paper. Additional assumptions required for our contention analyses will be mentioned in Section 3.

2.2 Deriving the Base Network Latency

The latency through the network without considering contention (T_b) is simply the product of the time through one node, and the sum of the nodes and the message length. The time through one node is the clock cycle time, which is the the sum of the switch and wire delay. That is,

$$T_b = (T^s + T^w(n)) \left(hops(n) + \frac{L}{W} \right)$$

As a base time period, let us denote the delay of a wire in a two dimensional network as $T^w(2)$. Let the switch delay T^s be greater than this wire speed by some constant factor s. Then,

$$T_b = T^w(2) \left(s + \frac{T^w(n)}{T^w(2)} \right) \left(hops(n) + \frac{L}{W} \right)$$

With randomly chosen message destinations, the average distance (k_d) a message must travel in each dimension in a network with unidirectional channels and end-around connections is given by

$$k_d = \frac{k-1}{2} \tag{1}$$

Therefore, for an n-dimensional network, $hops(n) = n(k-1)/2$. If there are bidirectional channels in a network with end-around connections, k_d is $k/4$ when k is even, and $(k - 1/k)/4$ when k is odd. The average distance in a dimension is $(k - 1/k)/3$ when the end-around connections do not exist.

We determine the length of the longest wire from an embedding of the n-dimensional network in a plane. The mapping is achieved by embedding $n/2$ dimensions of the network in each of two physical dimensions. Each added dimension of the network increases the number of nodes in the network by a factor k (recall $N = k^n$), and contributes to a $\sqrt{k}$ factor increase in the number of nodes in each physical dimension of space. If the distance between the centers of physically-adjacent nodes remains fixed, each additional dimension also increases the length of the longest wire by a $\sqrt{k}$ factor. (In practice, if the wire widths are non-negligible, the distance between adjacently-placed nodes nodes may also increase in high dimensional networks). Let the length of wires in a two-dimensional mesh $T^w(2)$ be measured as the distance between the centers of adjacent nodes. Then the length of the longest wire relative to the wire length in a two-dimensional network is given by

$$\frac{T^w(n)}{T^w(2)} = \sqrt{k}^{n-2} = k^{\frac{n}{2}-1}$$

The corresponding wire length for implementing the network in z-dimensional space is $k^{\frac{n}{z}-1}$. For example, given unit length wires in two-dimensional networks mapped onto a plane, the length of the longest wire in three-dimensional networks is $\sqrt{k}$. As stated before, this length determines the frequency at which packets can be sent over the wires. We note that the influence of long wires on the clock can be mitigated by allowing multiple clocks for transmission on long wires, or by allowing multiple bits to be in flight on the wire at any given time.

Substituting for the wire length, we now have

$$T_b = T^w(2)\left(s + k^{\frac{n}{2}-1}\right)\left(n\frac{k-1}{2} + \frac{L}{W}\right)$$

Replacing k with $N^{1/n}$, the latency equation becomes

$$T_b = T^w(2)\left(s + N^{\frac{1}{2}-\frac{1}{n}}\right)\left(n\frac{N^{\frac{1}{n}}-1}{2} + \frac{L}{W}\right) \tag{2}$$

In the above equation the channel width is chiefly affected when constraints on the bisection width or node size are applied. Our results will be normalized to a wire delay, $T^w(2)$, of 1.

3 A Contention Model for Buffered Direct Networks

This section derives a contention model for high-radix direct networks and validates it through simulations. The derivation proceeds like the buffered-indirect-network analysis of Kruskal and Snir [16]. Our contention model assumes buffered networks as well. Simulation experiments by Kruskal and Snir show that as few as four packet buffers at each switch node can approach infinite buffer performance, with uniform traffic. A buffer is associated with each output port of a switching node. If multiple packets request a given output port in a cycle, then we assume all but one packet are queued in the output buffer. We will derive network latency as a function of the channel utilization in the network.

Let us first derive an expression for the delay in a switching node with unit sized packets, and then extend the analysis to include larger packets. In an n-dimensional direct network, each switch has n network inputs and n network outputs, and a port leading to the processor connected to the node. The queue corresponding to each network output port can be treated as a queueing server, with v_i packets joining the queue during a cycle i. v_i is a random variable that can take on values ranging from 0 through $n+1$, corresponding to the n channels from the network and one from the processor. The v_i for different values of i are assumed to be independent random variables; let their expectation be E and variance be V. (In future, we will drop the use of the subscript on v.) E is the expected number of arrivals in any given cycle. As shown in [16], the average waiting time w for a packet in such a unit cycle-time system can be derived from the set of equations that result from an $M/G/1$ queueing system [15], as

$$w = \frac{V}{2E(1-E)} - \frac{1}{2} \tag{3}$$

3.1 Deriving the Distribution of v

To compute E and V we need the distribution of the random variable v. In an indirect network v has a simple binomial distribution because a packet from any input channel is steered towards an output queue with equal probability. In low dimensional, high-radix, direct networks that route packets completely in one dimension before the next, this is not the case. (See [28] for such a routing scheme. This routing scheme is commonly used in contemporary direct networks. We will comment on the performance tradeoffs in routing schemes that generate a more uniform distribution.) In such routing methods, the packets have a high probability of continuing in the same dimension when the network radix k is large yielding a more complicated distribution for v.

The distribution of v depends on k_d, the average number of hops taken by a packet in a dimension. Recall that for a network with unidirectional channels and end-around connections, $k_d = (k-1)/2$. For a network with a high value of k_d, a packet at an entering channel in a switch will choose to continue along an outbound channel in the same dimension with a high probability. Similarly, a packet at an entering channel in a switch will tend to change dimensions with a low probability; the lower the dimension of an outgoing channel, the lower the probability a message from a given high dimension will switch to that channel [7, 1].

We assume the routing probability of an incoming packet at a channel in a given dimension is non-negligible only for the continuing channel in that dimension, and for the channel corresponding to one lower dimension. We will also assume that incoming packets from the processing node are steered randomly to one of the output ports. In other words our analysis ignores the contribution to contention at an output channel of a switch of all but two incoming *network* channels – one incoming channel corresponding to the same dimension as the output channel, and the other incoming channel from one higher dimension – and the processor port. Making the above assumption allows us to obtain a simple closed form expression for the distribution of packets joining a queue. We will analyze these assumptions through detailed simulations in Section 3.3, where we will find that the assumptions yield accurate statistics even when k is small and n is large.

Let the probability of a packet arriving at an incoming channel be ρ (which is the channel utilization given our initial assumption of unit-sized packet). We can determine ρ as follows. Let the probability of a network request on any given cycle from a processor be m. The packet must travel k_d hops in each of n dimensions on the average, for a total of nk_d hops. Because each switch has n associated channels, the channel capacity consumed, or the channel utilization, is given by

$$\rho = \frac{mnk_d}{n} = mk_d \tag{4}$$

For networks with separate channels in both directions $\rho = \frac{mk_d}{2}$. The network bandwidth per node, or the message rate for which the network reaches saturation is obtained by setting $\rho = 1$.

To model contention in the switch, the packet probability ρ in a channel along a given dimension is composed of three components:

ρ_c: packets continuing along the same dimension through the switch

ρ_s: packets that switched to this dimension in the switch

ρ_i: packets injected into this dimension from the processing node at the switch (ρ_i packets on average from each channel also exit the network at a switch)

These switching probabilities are depicted in Figure 1 and are computed as follows. The probability a packet is generated by the processing node attached to the switch in any cycle is m, and the probability this packet routes to a given output channel in the switch is $1/n$, yielding $\rho_i = m/n = \rho/nk_d$.

Because the probability a packet exits the network from a channel is ρ_i, the probability it stays in the network is $\rho - \rho_i$. Since a packet switches dimensions once every k_d hops on average, the probability it will switch to one lower dimension in any given cycle is $\rho_s = (\rho - \rho_i)/k_d$.

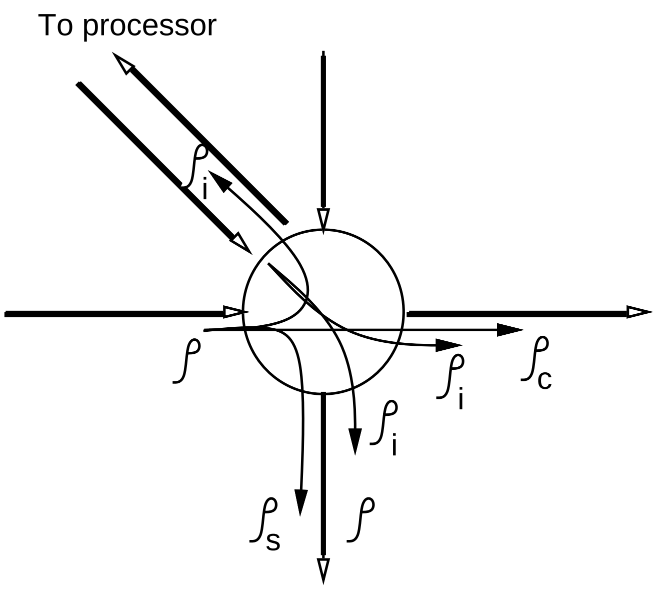


Figure 1: Channel utilizations at a switch. ρ is the probability of a packet at an input port of the switch from the network, ρ_c is the probability of continuing along the same dimension, ρ_s is the probability of switching dimensions, and ρ_i is the probability of entering the network into this channel from the processing node.

Similarly, its probability of continuing along the same dimension is $(\rho - \rho_i)(1 - 1/k_d)$, or, $\rho_c = (\rho - \rho_i)(k_d - 1)/k_d$.

We can now write the distribution of v as

$$p(v) = \begin{cases} (1-\rho_s)(1-\rho_c)(1-\rho_i) & v = 0 \\ \rho_c(1-\rho_s)(1-\rho_i) + \rho_s(1-\rho_c)(1-\rho_i) + \rho_i(1-\rho_c)(1-\rho_s) & v = 1 \\ \rho_c\rho_s(1-\rho_i) + \rho_s\rho_i(1-\rho_c) + \rho_i\rho_c(1-\rho_s) & v = 2 \\ \rho_c\rho_s\rho_i & v = 3 \\ 0 & v > 3 \end{cases}$$

Note that the above distribution yields $p(v) = 0$ for $v > 3$ in networks with three or more dimensions because of our assumption that only two dimensions contribute to contention. The distribution including the contribution due to all the channels could also be derived along the same vein, but the analysis would be more complicated.

The expectation and variance of the number of packets joining a queue are

$$\begin{aligned} E &= \rho_c + \rho_s + \rho_i = \rho \\ V &= \rho + 2\rho_c\rho_s + 2\rho_s\rho_i + 2\rho_i\rho_c - \rho^2 \\ &\approx \rho - \rho^2 + 2\rho^2\frac{(k_d - 1)}{k_d^2}\left(1 + \frac{1}{n}\right) \end{aligned} \qquad (5)$$

In the above equation for V we ignore higher order terms in $1/k_d$ in the expression within the rightmost parentheses. (As we see in our validation experiments, neglecting these terms does not appreciably change the results even for networks with a small radix, say $k = 4$.) Substituting the expressions for E and V in Equation 3 we get the average delay cycles through a switch

$$w = \frac{\rho}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right) \tag{6}$$

It is useful to note that the $1/n$ term corresponds to the contention arising from the message component from the processing node.

3.2 Including the Effect of Packet Size

We now extend the model to include non unit-sized packets. The effect of increasing the packet size to B flits can be approximated by increasing the delay through the switch by a factor B to reflect the increase in the service time of each packet, as would be the case in a system with synchronized message arrivals at a switch. Kruskal and Snir [16] made the same approximation in their analysis of buffered, indirect networks. We experimented with a wide range of networks and packet sizes to assess the validity of this approximation and found that it is indeed justified. (For example, the results of simulations with packet sizes two through 12 are depicted in Figure 5.) With B-flit packets, the channel utilization also increases by a factor B, yielding

$$\rho = mBk_d \tag{7}$$

The contention delay through a switch is

$$w = \frac{\rho B}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right) \tag{8}$$

and when $k_d >> 1$ further simplifies to

$$w = \frac{\rho B}{(1-\rho)} \frac{n+1}{nk_d} \tag{9}$$

The average transit time T_c through the network taking into account contention can now be computed in terms of the channel utilization. For pipelined switches (switches that use cut-through routing) with single cycle transit time through a switch, the average delay through a switch is thus $(1+w)$. Given that the total number of network links traversed is nk_d, we have

$$T_c = \left[1 + \frac{\rho B}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right)\right] nk_d + B \tag{10}$$

In the above equation, $nk_d + B$ is the minimum latency suffered by a request. As stated previously, $k_d = (k-1)/2$. The message size in flits is $B = L/W$, where L is message length in bits and W is channel width. Note that if the switch has a pipeline latency of p cycles, the switch delay will be p plus the contention component. When $k_d >> 1$, the transit time including contention is approximately

$$T_c = nk_d + B + \frac{\rho B(n+1)}{(1-\rho)} \tag{11}$$

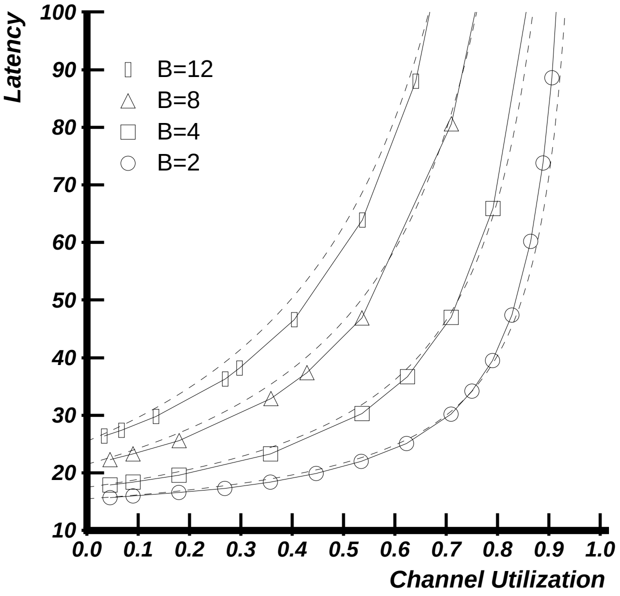


Figure 2: Comparing the direct network model with simulations for $n = 3$ and $k = 10$. Dashed lines correspond to model predictions.

3.3 Validation of the Model

We validated the model through simulations against several network types with a variety of workload parameters. In any given cycle, each node in the simulator generates a packet of size B with probability m to a randomly chosen destination node. The simulator routes messages through the network in the order of decreasing dimensions, and generates statistics such as average channel utilization in each dimension, average message latency, and observed message rates. Each simulation was run until the network reached steady state, that is, until a further increase in simulated network cycles did not change the measured channel utilization appreciably.

Figure 2 compares the network latency predicted by our model (Equation 10), and through simulations for a 1K-node network with $k = 10$ and $n = 3$ for several packet sizes. Figure 3 displays corresponding results for a two dimensional 100 node network with $k = 10$. We can see that our simple model predicts network latency with remarkable accuracy even at high loads.

The model overestimates the latency for large packet sizes, but underestimates it slightly at high loads. Let us examine the causes of these discrepancies. We believe the packet-size-related errors are due to the assumption of synchronized message arrivals at a switch. The difference at high loads can be attributed to the routing preference of messages injected into the network by the processing nodes. Our network simulator routes packets highest-dimension first in decreasing order of dimensions (see [28] for such a routing scheme). Such an ordered routing scheme allows deadlock-free routing in networks with finite-size buffers without end-around connections, and our general purpose simulator employs the same routing order for other networks as well. We verified in the simulations that the packets suffer higher-than-average delays in the higher dimensions.

We also modeled networks with separate channels in both directions without end-around connections. For such networks, $k_d = (k - 1/k)/3$, and $\rho = mBk_d/2$. The model's predictions were accurate for low to moderate loads, but the simulation latencies were higher than those of

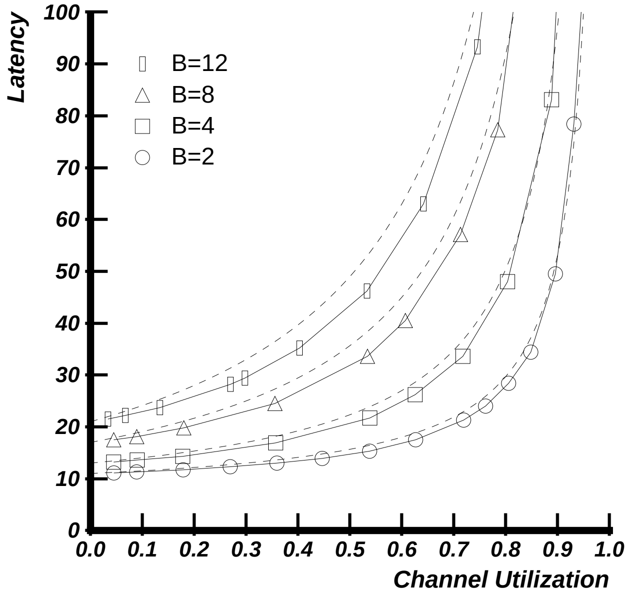


Figure 3: Comparing the direct network model with simulations for $n = 2$ and $k = 10$. Dashed lines correspond to model predictions.

the model when the channels were heavily loaded. The higher simulated latencies result from the non-uniform channel loading within each dimension: the missing end-around connections caused higher channel loads in the middle regions of each dimension.

It is interesting to see that routing packets in the order of decreasing dimensions causes uneven probabilities of packet arrivals from various dimensions to an output switch channel, and results in low contention until a high network utilization (notice the $1/k_d$ factor in the expression for the contention in Equation 9). This effect becomes more significant as k_d increases. This uneven probability results in a low value for the variance V in Equation 3. Note that V achieves its maximum value when $\rho_c = \rho_s$. The skew in the arrival distribution increases with network radix. Similar results were observed in buffered, binary n-cube networks by Abraham and Padmanabhan [1] and in unbuffered, direct networks by Dally [7].

We experimented with the model's predictive capabilities for low values of the radix k. Comparisons of the model and simulations for networks with $k = 4$ and $n = 2$, 3, and 4, are shown in Figure 4. In general, we see that the model is robust; its predictions remain accurate even for low values of k, although as before, the model underestimates latency as the network approaches saturation. (The complete expression for V in Equation 5 yields virtually the same curves.) The simulation curves are higher than those of the model at high loads in Figure 4 than in Figures 3 and 2; this is an artifact of non-negligible switching probabilities from several lower dimensions when k is small. These switching probabilities could be safely ignored at high values of k.

Figure 5 illustrates how packet size affects network contention. We derive network latency as a function of packet size, given a fixed number of bits transferred (mB), for several values of n and k. The request rates m are adjusted to achieve the same throughput ($\rho = 0.4$) for various packet sizes. For example, the request rate needed to achieve a 0.4 channel utilization for packet size B is given by $m = 0.4/(Bk_d)$. It is clear from the model (and confirmed by our simulations) that increasing packet size (for the same number of bits transferred) results in a

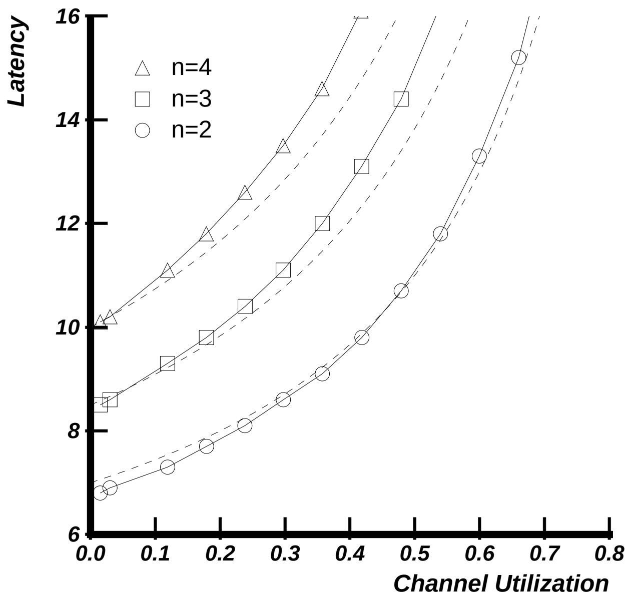


Figure 4: Comparing the direct network model with simulations for a small values of the radix. Radix $k = 4$, $B = 4$, and $n = 2$, 3, and 4. Solid lines are measurements from a simulator, dashed lines correspond to predictions by the model.

proportional increase in network contention. The good correspondence between the model and simulations for the various network types validates our packet size extension to the contention model. As stated before, the model's overestimation of latency for large packets results from our assumption of simultaneous message arrivals at a switch.

We also see that the network can be operated without significant loss of performance at higher capacities when the packet size is small (see Figures 2 and 3). That is, smaller packets allow the network to be operated closer to the network's theoretical peak bandwidth, without significant degradation in latency. Conversely, for a given channel utilization, smaller messages result in lower latency (see Figure 5). Kruskal and Snir's indirect network model predicted a similar dependence of latency on blocksize, prompting Norton and Pfister to consider splitting messages into multiple smaller ones in the RP3 [21]. However, splitting long messages into multiple smaller ones with back to back transmission may not realize the higher predicted throughput because of the destination correlation of these sub-messages and the relatively high packet overhead. The node must also be able to support multiple outstanding sub-messages in the network. We believe higher throughput might be achieved with splitting if some delay is introduced between sending the sub-messages at the source, or by randomizing the routes taken by the sub-messages. We are currently investigating the potential performance benefits of such methods.

The contention, switch, and wire delay components of the latency (from Equations 2 and 10) can be combined to yield the effective network delay:

$$T_c = T^w(2)\left(s + N^{\frac{1}{2}-\frac{1}{n}}\right)\left[nk_d\left(1+w\right) + \frac{L}{W}\right] \tag{12}$$

where w is contention delay per node.

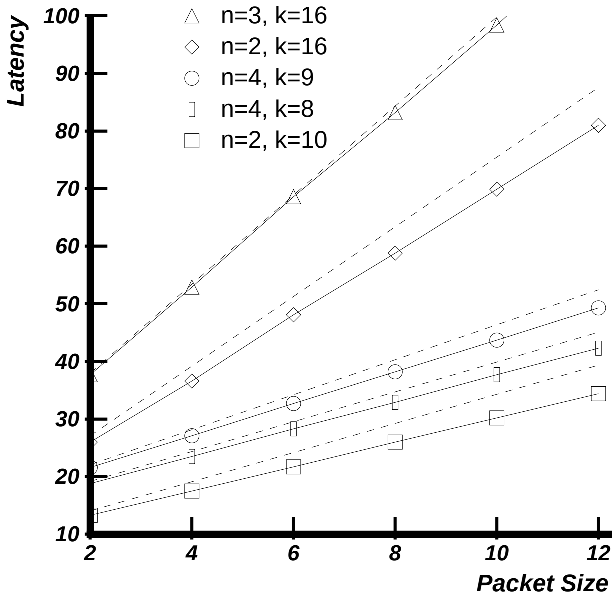


Figure 5: Assessing the effect of packet size. Dashed lines correspond to model predictions. Channels are operated at a utilization of 0.4.

4 Analysis of Base Network Latency

The next three subsections analyze network performance using the base network latency (ignoring contention) with different constraints. To enable a comparison, Appendix A presents network latency results when node delays are ignored and bisection width is fixed. An analysis of networks including contention effects and communication locality will follow here. Our results assume $T^w(2)$ is normalized to 1.

4.1 Keeping Channel Widths Fixed

The base network latency T_b is derived using Equation 2. Let us start with no constraints on the bisection width or node size to allow a constant channel width over networks of all dimensions (i.e., W is held constant in Equation 2). Keeping channel widths fixed allows boundless system size increase for high dimensions and is impractical when the number of processors is large, but it allows us to reflect on fundamental constraints imposed by signal propagation delays alone.

We plot latency as a function of network dimension in Figure 6 for several system sizes if switch delays also contribute to latency. Message length L/W is assumed to be 20 flits. Note that the Y axes in the graphs use a logarithmic scale and small relative differences in the curves can be significant. The graph in Figure 6 shows that for 16K and 1M nodes, a three-dimensional topology is superior to a two-dimensional topology. This is in contrast to previous results in [7] where the linear wire delay model yielded the lowest latency for two-dimensional networks when the switch delay was ignored and the bisection width was held constant.

A more important point is that physical limitations favor low dimensional networks, even when bisection width is unconstrained, because of the rapid increase in wire length as n is increased. If the number of dimensions is increased from n to $n+1$ for the same total number of processors N, the wire length increases by a factor of $N^{\frac{1}{n(n+1)}}$.

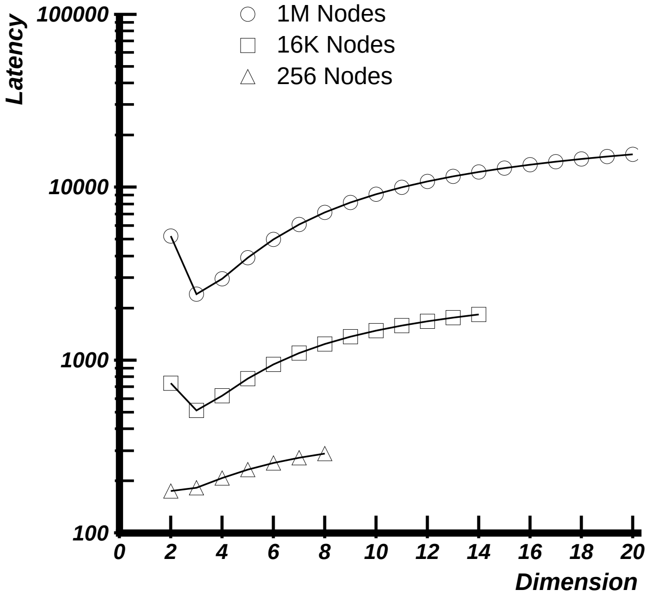


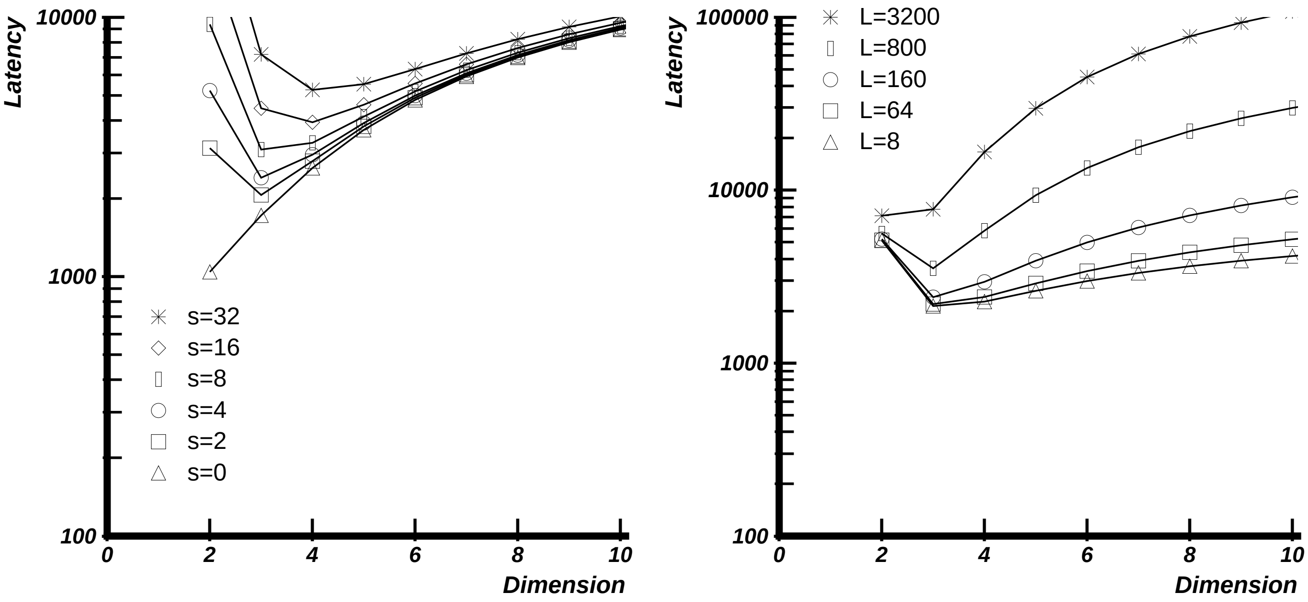
Figure 6: Latency for systems with 256, 16K, and 1M nodes assuming node to base wire delay ratio to be 4.0. Channel widths W are fixed at 8 bits. Message lengths (L/W) are assumed to be 20 flits.

Figure 7(a) evaluates the effect of relative switch delay on network latency for 1M-node systems. As expected, the dimension of the best latency network shifts towards greater dimensions as the switch delay increases in proportion to the base wire delay. With zero switch delay, the two-dimensional network is indeed the best choice for message length L/W equal to 20 flits, with switch delays 2 through 8, the three-dimensional network is optimal, while a relative switch delay of 16 suggests that four dimensions are best.

Figure 7(b) shows that larger message lengths favor networks with a lower dimensionality. The reason is that the number of hops in the latency equation becomes dominated by the message traversal time for long messages, making the gains because of fewer hops in a high-dimensional network less useful. Only when messages are exceedingly long (say, 3200 bits) does a two-dimensional network have the lowest latency.

The effect of switch delay and message length on networks with 1K nodes is shown in Figure 8. For a fewer number of processors, the effect of wire delays is less pronounced when switch delays are greater than four times the wire delay. In this case, three-dimensional networks become favorable only when switch delays exceed 4. Most practical message lengths favor three dimensional networks.

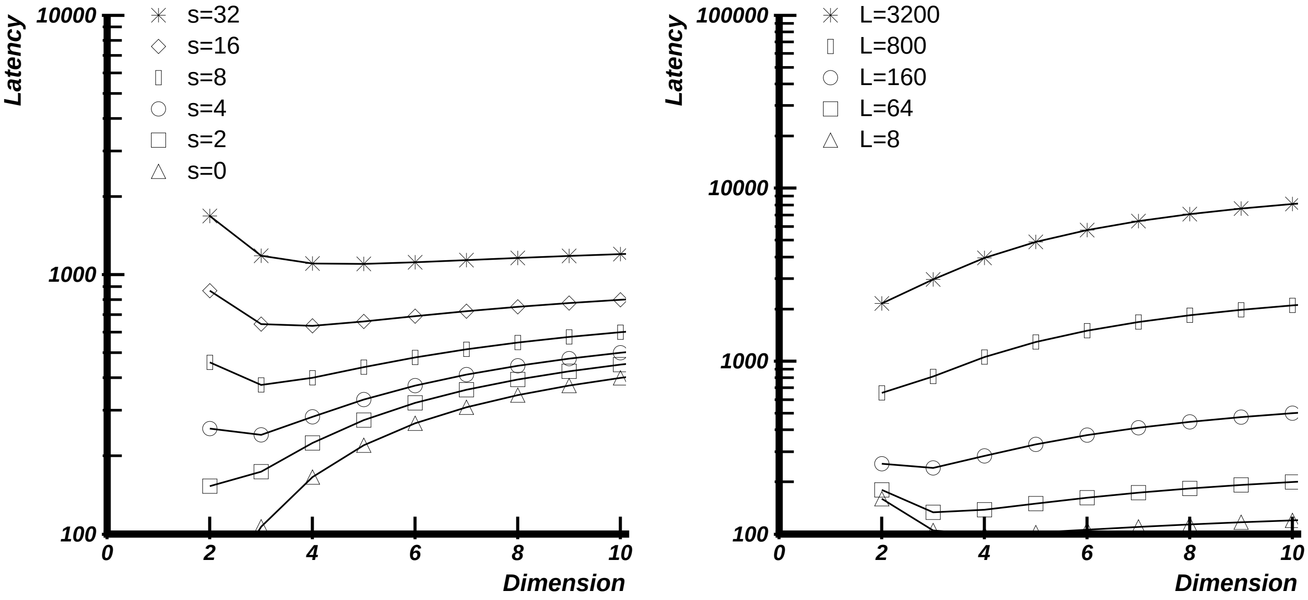
The analysis presented in this section made several different assumptions from those made in previous studies. First, we assumed that the bisection width is not a constraint. Second, we separately included switch delays and wire delays. Third, we explored the effect of smaller message sizes. Allowing greater bisection widths for networks with higher dimensions allowed the channel widths to remain the same for all network topologies. Modeling switch delays separately made the effect of longer wires less severe. Finally, we saw that smaller messages make the number of network hops traveled more significant in comparison with the pipeline delay of a



(a) Latency for various node delays

(b) Latency for various message lengths

Figure 7: Latency for 1M-node systems fixing only channel widths. (a) Varying node delays. Node to base wire delay ratio s ranging from 0 through 32. Message lengths (L/W) are assumed to be 20 flits. (b) Varying message lengths, with $s = 4.0$. Message lengths L/W range from 1 through 400 flits ($W = 8$).



(a) Latency for various node delays

(b) Latency for various message lengths

Figure 8: Latency for 1K-node systems fixing only channel widths. (a) Varying node delays. Node to base wire delay ratio s ranging from 0 through 32. Message lengths (L/W) are assumed to be 20 flits. (b) Varying message lengths, with $s = 4.0$. Message lengths L/W range from 1 through 400 flits ($W = 8$).

message. These inclusions tend to favor the choice of networks with higher dimensionality.

4.2 Keeping Bisection Width Fixed

In general, the bisection width cannot be increased arbitrarily. Because this width imposes bounds on the minimum layout area [29], allowable system size, switch node size, cost, and power considerations will limit this width. In such cases, the bisection width can be held constant at some limit, which directly affects the channel width and hence the average message length. The optimal value of n can then be derived.

For a fixed bisection size, channel widths become smaller with increasing n. We must also point out that if bisection width is a design constraint, then the *wire widths* must surely be significant. For now, we do not model the increase in wire lengths as n is increased due to non-negligible wire widths; we will touch on this issue later in this section.

The bisection width can be easily computed for a network with n dimensions and radix k. Adding an extra dimension to a network with $n-1$ dimensions requires the addition of an extra channel to each of the k^{n-1} nodes, which results in k^{n-1} channels in the highest dimension in the new n-dimensional network. These channels contribute to the bisection width. Thus, the bisection width for a k-ary n-cube with W-bit channels is $2Wk^{n-1}$ (the factor 2 accounts for the end-around channels). If N is the number of nodes, the bisection width is $2WN/k$. For example, in a linear array of processors, the bisection width is $2W$. The bisection width becomes $2Wk$ when each row of processors is replicated k times, and becomes $2Wk^2$ when this 2-D array is replicated k times for a 3-D network.

We will normalize bisection widths to that of a binary n-cube with unit-width channels. The bisection width of a binary n-cube with unit-width channels is N; consequently, the channel width W of a k-ary n-cube is derived as $k/2$. For example, in a 256 node system, if the binary 8-cube has unit-width channels, the corresponding mesh network ($n = 2$) has $W = 8$.

The normalization to the unit-channel-width binary n-cube is rather arbitrary, but it is easy to carry out a similar analysis using a different bisection width, obtained using some realistic cost constraint.

We derive the corresponding latency by substituting $W = k/2 = N^{\frac{1}{n}}/2$ in Equation 2, as

$$T_b = T^w(2)\left(s + N^{\frac{1}{2}-\frac{1}{n}}\right)\left(n\frac{N^{\frac{1}{n}}-1}{2} + 2\frac{L}{N^{1/n}}\right)$$

Figure 9 shows the latencies as a function of dimensions for various system sizes. The chief difference between the results here and those when the bisection is unconstrained is that the latency for large values of n is much larger, which results from smaller channel widths in addition to longer wire delays. As mentioned earlier, adding a network dimension with the same total number of nodes increases wire delays by a factor of $N^{\frac{1}{n(n+1)}}$, and increases message length in flits by the same factor. This result makes a stronger case for low-dimensional networks when bisection width is fixed.

The relative shapes of the curves for a 1M-node system with varying switch speed and message length (not shown here) are similar to those in Figure 7. The primary result is that the network dimension that yields the minimum latency for various system sizes, switch-to-wire delays, and message lengths, is generally the same as the best dimension when the bisection width is not constrained.

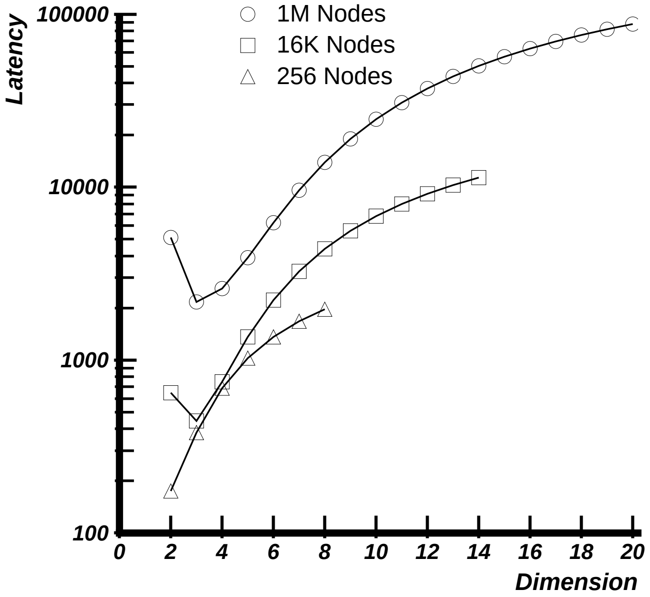


Figure 9: Latency for systems with 256, 16K, and 1M nodes assuming node to base wire delay ratio s to be 4.0. Message lengths (L) are assumed to be 160 bits. Bisection is normalized to a binary n-cube with unit-width channels.

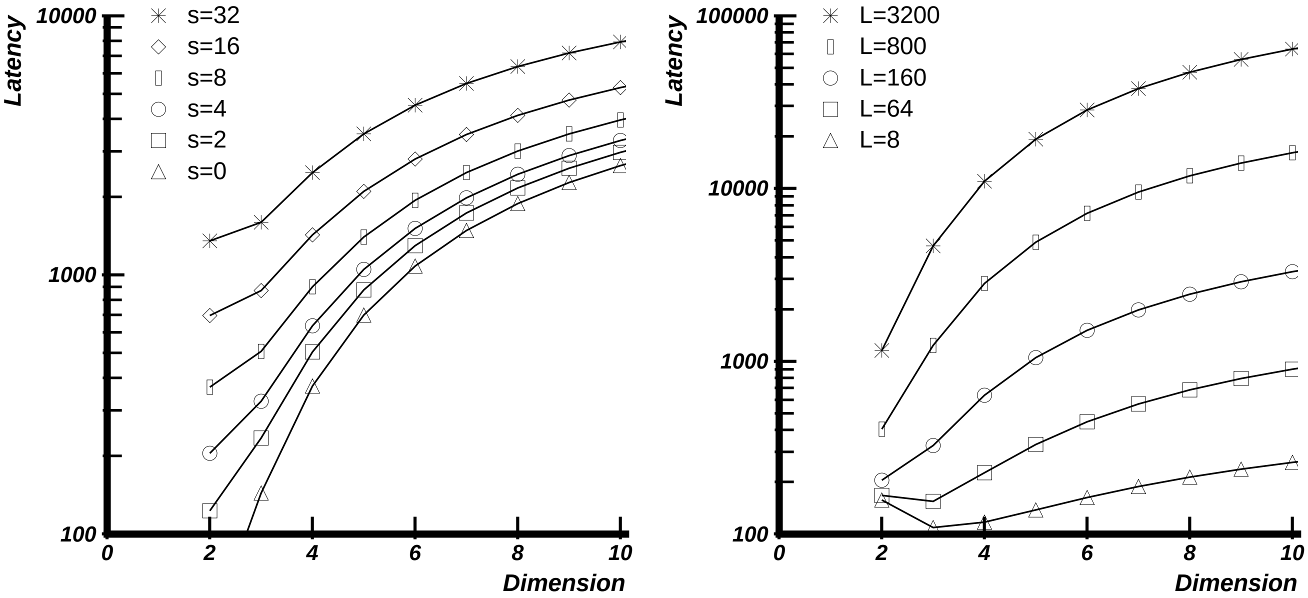
The tradeoffs are slightly different in smaller systems. Figures 10(a) and 10(b) analyze latency for a 1K-node system with different switch-to-wire delays and message lengths respectively. When bisection width is fixed, we see that the two-dimensional network is generally the best. In 1K-node systems, the relative impact of longer messages caused by narrowing channels is more significant than reducing the number of hops.

Our assumption in Section 2.2 of a fixed distance between physically-adjacent nodes implies that decreasing W does not impact either the length of the longest wire or the switch delay. Alternatively, we can assume that the distance between adjacent nodes is proportional to the channel width W. In this context, let our normalization factor, $T^w(2)$, represent the wire length in a unit-channel-width 2-D mesh network. Therefore, in a k-ary n-cube, the delay due to the longest wire is proportional to

$$WN^{\frac{1}{2}-\frac{1}{n}}.$$

Interestingly, with the above dependence on W, the length of the longest wire in a bisection-constrained network turns out to be a constant for all n. With the bisection normalized to a binary n-cube with $W = 1$, the channel width can be written as $W = N^{\frac{1}{n}}/2$. The resulting wire delay is proportional to $N^{\frac{1}{2}}/2$, which is a constant for all n.

In this analysis, because bisection width is normalized to a binary n-cube, the channel widths of smaller-dimensional networks appear to be impractically large. Node size limitations will often be the constraining factor, and an analysis using pin-limited nodes is presented next.



(a) Latency for various node delays

(b) Latency for various message lengths

Figure 10: Latency for 1K-node systems fixing bisection width. Bisection width is normalized to a binary n-cube with unit channel width. (a) Varying node delays. Node to base wire delay ratio s ranging from 0 through 32. Message lengths (L) are fixed at 160 bits. (b) Varying message lengths, with $s = 4.0$. Message lengths range from 8 through 3200 bits.

4.3 Keeping Node Size Fixed

In practical systems, node sizes often constrain channel widths rather than bisection width. For example, pin limitations in VLSI switches allow channel widths of 8 or 16 bits for three-dimensional networks, with separate channels for each direction, or 16 or 32 bits without separate channels. This section fixes the total number of wires emanating from a switch. When n increases, for a fixed node size, the effective message length in flits increases because W must decrease.

Let us analyze networks with node sizes normalized to that of a 2-D mesh network with channel width $W = 32$. These widths yield 128 signal wires per node (not counting wires to the processor, power and ground lines) assuming unidirectional channels. Such a node can be integrated into a single VLSI chip with current technology. The channel width of a n-dimensional network is then $W = 64/n$. As before, the default message length is 160 bits. Substituting in Equation 2, the base network latency becomes

$$T_b = T^w(2)\left(s + N^{\frac{1}{2}-\frac{1}{n}}\right)\left(n\frac{N^{\frac{1}{n}}-1}{2} + \frac{Ln}{64}\right)$$

The latency curves for various system sizes are similar in nature to those in Figure 6 because W has a lower sensitivity to n, and because the normalization was done with respect to a 2-D mesh with $W = 32$. For the same reasons, the curves for various switch speeds and various message lengths are similar in nature to those in Figures 7 and 8 respectively. That is, as switch speeds are increased from $s = 2$ to $s = 32$, the optimal n shifts from 2 to 4, and as message lengths are increased from 8 bits to 3200 bits, the most desirable value of n shifts from 3 to 2.

5 Effect of Network Contention

This section explores the effect of the available network bandwidth in various networks using the contention model derived in Section 3. We use the following constraints:

- ignore wire delays, with constant channel widths
- include wire delays, with constant channel widths
- include wire delays, with fixed bisection width
- include wire delays, with constant node size.

We consider networks with $N = 1K$ nodes, and $n = 2$, 3, 4, and 5. We assume switch delay $s = 4$, message length $L = 128$, and normalize channel widths to a network with $n = 2$ and $W = 32$.

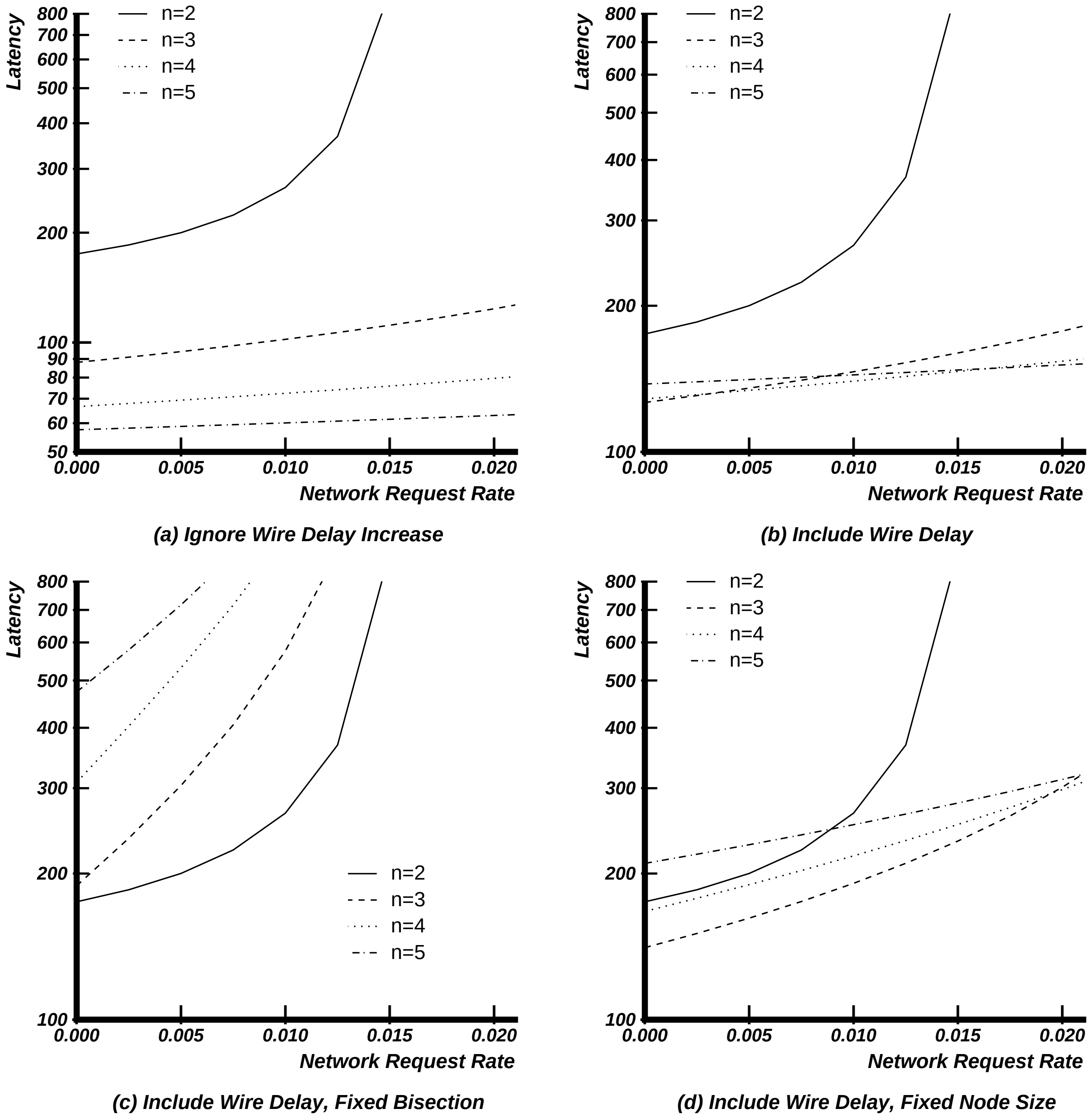
When channel widths are held constant, low-dimensional networks have a smaller capacity that networks with more dimensions. Recall that channel utilization $\rho = m\frac{L}{W}k_d$ (from Equation 7). The theoretical network-capacity limit is reached when the product of the request rate m and message size in flits L/W is $2/(k-1)$.

In general we found that the variability in network latency for different constraints is much greater than when network contention is ignored. Figure 11(a) shows the effective network latency when wire delay is ignored. As expected, the higher-dimensional networks have lower latencies for a given request rate. Clearly, the two-dimensional network lacks the bandwidth to support a request rate of more than about 0.015 (solving $mL/W = 2/(k-1)$, yields $m = 1/64$), while the three-dimensional network can support a request rate of up to about 0.05.

Figure 11(b) shows the corresponding network latency when wire delays are included. Here we see that the performance of higher-dimensional networks is significantly hurt by wire delay. For low request rates, when bandwidth requirements do not dominate, networks with $n = 2$, 3, 4, and 5, perform similarly and are clearly superior to the two-dimensional network. Our base latency analysis yields similar conclusions (see Figure 8).

Limiting the bisection width dramatically changes the relative standing of the low and high-dimensional networks. The constrained bisection results in narrower channels for higher dimensions, which not only increases the base latency, but also reduces the available network bandwidth. Figure 11(c) shows the network latency when bisection is constrained, and when switch delays and wire delays are included. We normalize the bisections to that of a two dimensional network with 32 bit channels. That is, $W = 32$ for $n = 2$ and $N = 1K$. This constraint is reflected in the value of the channel width W for the various networks. In Equation 12, with the above bisection constraint, $W = k = N^{\frac{1}{n}}$. Thus, for a five-dimensional network $W = 8$, which yields 16-flit messages for $L = 128$. Here we see that the higher-dimensional networks suffer much higher delays, and we obtain results similar to those in [8]. With fixed bisection, the two-dimensional network outperforms the rest.

Figure 11(d) plots network latency when the node size is fixed, and normalized to that of the two-dimensional network with $W = 32$. When node size is constrained, the two-dimensional network performs poorly at high loads (when $m > 0.01$) because it suffers significant contention; its performance is reasonable at low loads.



(a) Ignore Wire Delay Increase

(b) Include Wire Delay

(c) Include Wire Delay, Fixed Bisection

(d) Include Wire Delay, Fixed Node Size

Figure 11: Effect of contention on network latency with 1K nodes, $L = 128$, and $s = 4$. (a) Ignore increase in wire delay (assume unit wire delay), and $W = 32$. (b) Include wire delay, $W = 32$. (c) Include wire delay and fix bisection (normalize to network with $n = 2$ and $W = 32$). (d) Include wire delay and fix node size (normalize to network with $n = 2$ and $W = 32$).

5.1 Communication Locality

Direct networks can take advantage of communication locality in parallel applications. Informally, we say that communication locality exists when the likelihood of communication (or access) to various nodes decreases with distance. Packets destined for neighboring nodes not only travel fewer hops, but also consume a smaller fraction of the network bandwidth. This section analyzes the effect of communication locality on network throughput and latency.

Our model can be easily extended to account for communication locality using the following simple locality model. Let us define the *locality fraction* l as the fraction of all processors that are potential candidates to receive a message from a source node. Furthermore, for a given source node, let message destinations be chosen randomly from the n-dimensional subcube with $N \times l$ processors centered at the source node. For example, let us consider an N-processor torus in which nodes are represented by their x and y coordinates. Given a locality fraction l, destination nodes for messages originating from source node (i,j) are randomly chosen from the set of nodes with coordinates $(x| \; i \leq x \leq i + \sqrt{lN} - 1, \quad y| \; j \leq y \leq j + \sqrt{lN} - 1)$. (Other forms of communication locality could also be realized by using some probability function to represent higher than average access likelihoods to nearby nodes, or to favor straight through paths over paths that require turns).

With the above locality model, a packet travels an average distance of k_{dl} in each dimension, for a total of nk_{dl} hops from source to destination. The average distance traversed in a dimension can be expressed as

$$k_{dl} = ((lN)^{1/n} - 1)/2 = (l^{1/n}k - 1)/2$$

The average latency can be derived by replacing k_d in Equation 10 with k_{dl}. The same substitution is necessary in Equation 7 to compute ρ. Destinations chosen randomly over the entire machine corresponds to $l = 1$.

Locality increases the effective throughput and decreases the latency of the network. The network reaches full capacity when all channels are fully utilized, that is, when $\rho = mBk_d = 1$. (Although this ideal throughput is hard to achieve in practice owing to contention.) The peak network throughput in messages per cycle per node is $1/Bk_d$, and is $1/k_d$ in flits per cycle per node. However, when communication locality exists, the throughput increases to $1/k_{dl}$ flits per cycle per node. Similarly, the base network latency of $nk_d + B$ hops decreases to $nk_{dl} + B$ when locality exists. In other words, locality increases throughput by a factor $1/l^{1/n}$, and decreases base network latency by the same factor (when $nk_d >> B$).

Locality improves latency because it reduces both the number of hops per packet and average contention delays. As displayed in Figure 12, with a light load of $m = 0.001$, latency reduction is largely due to the fewer number of hops. At light loads, latency is linearly related to k_{dl} or to $l^{1/n}$, which is clear from Equation 10 when the contention component is ignored. For example, when $m = 0.001$, for a 1K-node machine ($n = 2$ and $k = 32$), the average latency for randomly selected destinations is roughly 35. When the average distance in a dimension decreases by 10% ($l^{1/2} = 0.9$), the latency decreases by the same fraction to 31.

The impact of locality is much more significant when contention is high. In this case the latency reduction due to locality is largely due to a reduction in the bandwidth requirement. The latency at high loads is proportional to $1/(1 - mBk_{dl})$. For example, the average latency drops by over 25% (from 67 to 50) for the higher load of $m = 0.012$, when $l^{1/2} = 0.9$. Of this decrease, over 19% is due to the reduced bandwidth consumed, while less than 6% is due to the fewer number of hops. Thus we see that communication locality has a much stronger effect on

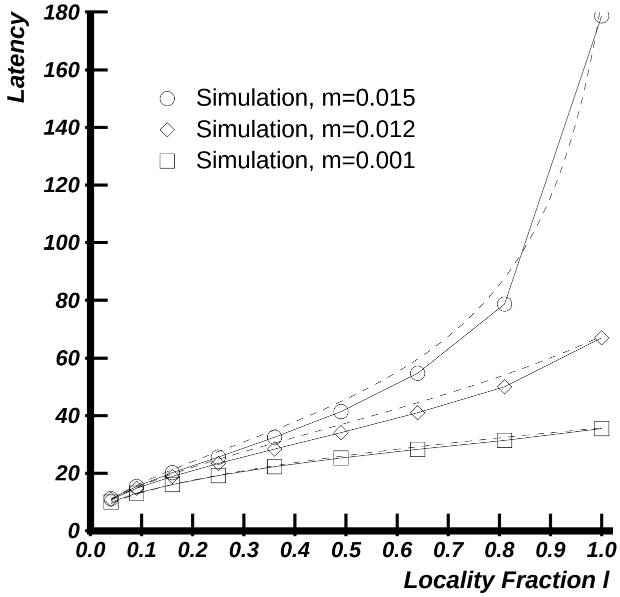


Figure 12: Effect of communication bandwidth and latency reduction due to locality, with $N = 1K$, $n = 2$, $k = 32$, and $B = 4$. The locality fraction, l, is the fraction of all processors that can potentially receive a message from a given node. Dashed lines correspond to model predictions and points are taken from a simulator.

network performance through its reduction of bandwidth consumed than through its reduction of base network latency. The proportional impact of locality is even more significant at higher loads.

Communication locality makes low-dimensional networks more competitive with other networks. Although low-dimensional networks have shorter wires and smaller bisections, their lower available bandwidth and higher base latencies reduce their effectiveness. Locality mitigates these negative aspects of low-dimensional networks by reducing the effective distance a message travels, consequently decreasing bandwidth requirements and the base latency. For example, compare the performance of the two-dimensional network relative to other networks in Figure 13, which assumes a communication locality fraction of $l = 0.3$, with that in Figure 11(d), which assumes no locality. We see that communication locality has a larger relative effect on the two-dimensional network.

5.2 Direct Versus Indirect Networks

In the past, shared-memory multiprocessors (e.g., the Ultracomputer [11], RP3 [23], Cedar [10], and BBN Butterfly) have generally employed indirect networks. These networks provide uniform-cost access to remote memory modules, and have a high bandwidth, but they do not allow the exploitation of locality in the communication patterns of parallel applications. Because they can exploit locality, distributed shared-memory multiprocessors based on direct networks can scale to a large number of processors for computations that exhibit locality.

This section compares the latency of direct networks with that of indirect networks under various constraints and workload conditions, for varying degrees of communication locality. Indirect networks for N processors are made up of n stages of $k \times k$ crossbar switches, where

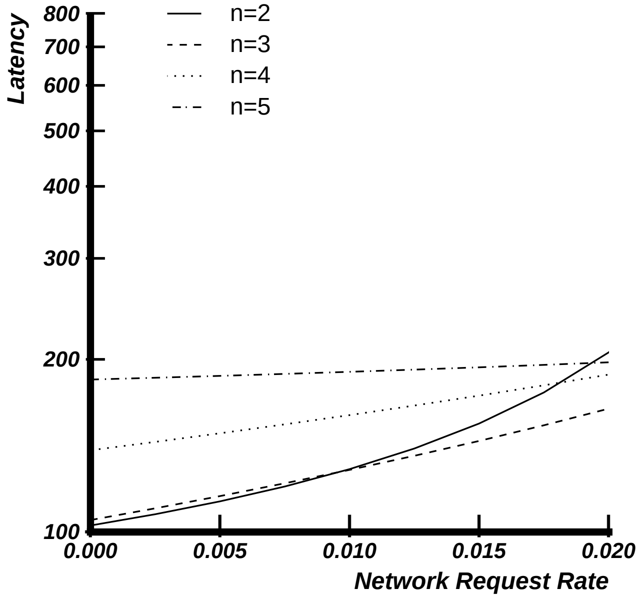


Figure 13: Assessing the relative effect of communication locality on networks with various dimensions. Communication locality parameter $l = 0.3$, $N = 1K$ nodes, $L = 128$ bits, and $s = 4$. Include wire delay and fix node size (normalize to network with $n = 2$ and $W = 32$).

$N = k^n$. The indirect network forces each request to travel the n stages to its destination.

We will compare the latency of direct and indirect networks under a node-size constraint. The length of the longest wire in a planar mapping of an indirect network can be derived by noting that the indirect binary n-cube network is isomorphic to the direct binary n-cube [27]. The latency for an n-stage indirect network with $k \times k$ switches can be written as

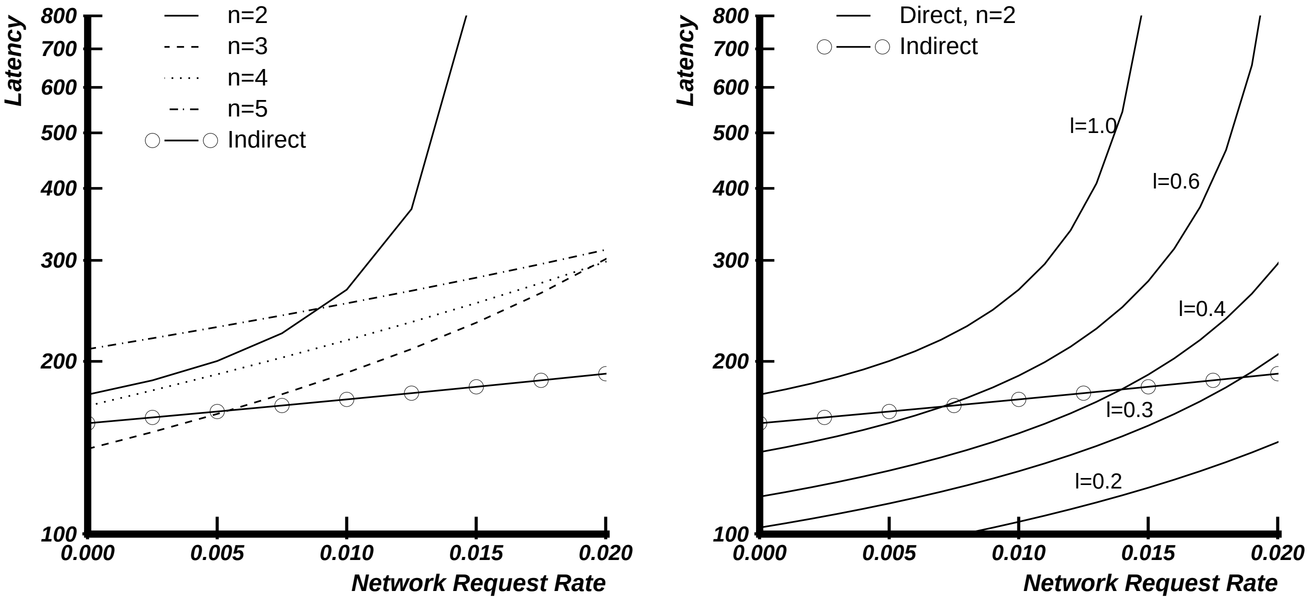
$$T_c = \left[n\left(1 + w\right) + \frac{L}{W} \right] \tag{13}$$

Kruskal and Snir [16] derived the contention delay w per switch stage for buffered indirect networks as[1]

$$w = \frac{\rho B}{2(1-\rho)} \left(1 - \frac{1}{k}\right) \tag{14}$$

where the message length in flits $B = L/W$ and the channel utilization $\rho = mB$. The derivation of the contention component in indirect networks differs from direct networks in the computation of $p(v)$, which is the distribution of the number of packets joining an output queue of a switch (see Section 3). Because packets from k input ports in the indirect network are steered towards an output queue with equal probability, the distribution of $p(v)$ is binomial and results in a much simpler form of w (compare with the expression for w for direct networks in Equation 8). Kruskal and Snir have validated this model for various networks and message rates; we verified the accuracy of this simple model for several packet sizes by comparing its predictions with simulations.

[1] Kruskal, Snir, and Weiss derive more accurate formulas in [17], but this is sufficient for our purposes.



(a)

(b)

Figure 14: Comparing direct network latency with indirect networks for 1K processors and message length $L = 128$. Switching node sizes normalized to direct network with $n = 2$ and $W = 32$. (a) No communication locality. (b) With communication locality, $n = 2$.

Let us analyze the latency of 1K-processor networks with $L = 128$ bits. Figure 14(a) compares the latency of indirect networks with several direct networks as a function of network load. In this graph, the destinations are assumed to be randomly chosen. With 4×4 switches, ($k = 4$ and $n = 5$) the 1K-node indirect network has roughly the same number of switches as the direct network (1280 versus 1024) making a comparison based on pin constraints meaningful. The number of pins per switch is fixed at 128. Therefore, the torus network has $W = 32$ $(128/2n)$, and the indirect network has $W = 16$ $(128/2k)$.

When locality is not taken into account, the indirect network has the lowest latency and the highest bandwidth of all the networks shown. A three-dimensional direct network reaches saturation when the message rate $m = 2/B(k-1) = 0.037$ messages per cycle; the indirect network has a saturation message rate of $m = 1/B = 0.125$.

Direct networks can take advantage of communication locality. Figure 14(b) compares the performance of the indirect network with a two-dimensional direct network for varying degrees of locality. Recall that locality reduces the number of potential destination nodes for a given source node by a fraction l. The figure shows that the latencies of the two networks is comparable for low message rates when $l < 0.6$, and at high loads when $l < 0.3$. Thus we see that communication locality can make low-dimensional direct networks perform as well as – or even better than – indirect networks, even though indirect networks have a greater bisection width.

6 Conclusions

The performance of multiprocessor interconnection networks is influenced by switch delays and wire delays. This paper analyzed the relative effect of switch and wire delays under various constraints such as fixed bisection width, fixed channel widths, and fixed node sizes. We derived a simple model for contention in buffered direct networks and assessed its impact on network performance for the above constraints.

Under the constraint of constant wire density and constant bisection width, previous results for network embedding in a plane showed that a two-dimensional mesh yields the lowest latency. However, when node delays are taken into account, we showed that the best network has a moderately high dimension.

Message length plays an important role in the tradeoff between low and high-dimensional networks. Longer messages (such as those in message passing machines) make the relative effect of network distance from source to destination less important, while the lower expected message lengths in shared memory machines increase the relative influence of network distance, and tend to favor networks with more dimensions.

We introduced a contention model for buffered, direct networks to estimate the effect of network bandwidth. We validated the model through simulations, and demonstrated its robustness over a wide range of radices and dimensions. We evaluated the performance of networks including contention with constraints such as fixed bisection width and fixed node size. An interesting finding of this analysis is that the relative standing of networks is strongly dependent on the constraints chosen and on the expected workload parameters. In contrast, the results showed much less variance when bandwidth considerations were ignored. While it is true that lower-dimensional networks have a lower bandwidth than higher-dimensional networks, the higher-dimensional networks suffer when wire delays are included and the bisection size is constrained. However, when the less stringent limit of a fixed node size was applied, the situation is completely reversed: at low loads three-dimensional networks have the lowest latency, and at high loads four-dimensional networks are superior.

Direct networks can exploit communication locality; locality improves both network throughput and latency. At low loads, communication latency decreases linearly with the average distance traversed in each dimension. The relative decrease in latency is even more significant when network load is high, owing to a reduction in the bandwidth requirements of the application.

Communication locality enhances the attractiveness of low-dimensional networks. Although low-dimensional networks have shorter wires and smaller bisection widths than other networks, their lower available bandwidth and higher base latencies reduce their effectiveness. Locality mitigates these negative aspects of low-dimensional networks by reducing the effective distance a message travels, consequently decreasing bandwidth requirements and the base latency.

Communication locality depends on several factors including the architecture of the multiprocessor, the compiler and runtime systems, and the characteristics of parallel applications. If the communication locality of parallel applications can be enhanced through better algorithms and systems architectures, parallel machines with significantly higher performance can be built without incurring the high cost of expensive networks.

7 Acknowledgments

My gratitude to Bill Dally, Ken MacKenzie, Gino Maa, Steve Ward, Tom Knight, Dan Nussbaum, Tom Leighton, and Charles Leiserson for all the interesting discussions that fueled this research. I would also like to thank the referees for their constructive suggestions. David Chaiken, Bill Dally, Beng Hong Lim, and Peter Nuth provided feedback on a draft of this paper.

Gino Maa wrote the Alewife network simulator and helped validate the network model.

References

[1] Seth Abraham and Krishnan Padmanabhan. Performance of the Direct Binary n-Cube Network for Multiprocessors. *IEEE Transactions on Computers*, 38(7):1000–1011, July 1989.

[2] Anant Agarwal, Beng-Hong Lim, David A. Kranz, and John Kubiatowicz. APRIL: A Processor Architecture for Multiprocessing. In *Proceedings 17th Annual International Symposium on Computer Architecture*, pages 104–114, June 1990.

[3] William C. Athas and Charles L. Seitz. Multicomputers: Message-Passing Concurrent Computers. *Computer*, 21(8):9–24, August 1988.

[4] Shekhar Borkar et al. iWarp: An Integrated Solution to High-Speed Parallel Computing. In *Proceedings of Supercomputing '88*, November 1988.

[5] David Chaiken, Craig Fields, Kiyoshi Kurihara, and Anant Agarwal. Directory-Based Cache-Coherence in Large-Scale Multiprocessors. *IEEE Computer*, 23(6):41–58, June 1990.

[6] David Chaiken, John Kubiatowicz, and Anant Agarwal. LimitLESS Directories: A Scalable Cache Coherence Scheme. In *Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IV)*, ACM, April 1991. To appear.

[7] William J. Dally. *A VLSI Architecture for Concurrent Data Structures*. Kluwer Academic Publishers, 1987.

[8] William J. Dally. Performance Analysis of k-ary n-cube Interconnection Networks. *IEEE Transactions on Computers*, 39(6):775–785, June 1990.

[9] William J. Dally et al. The J-Machine: A Fine-Grain Concurrent Computer. In *IFIP Congress*, 1989.

[10] Daniel Gajski, David Kuck, Duncan Lawrie, and Ahmed Saleh. Cedar – A Large Scale Multiprocessor. In *International Conference on Parallel Processing*, pages 524–529, August 1983.

[11] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph, and M. Snir. The NYU Ultracomputer – Designing a MIMD Shared-Memory Parallel Machine. *IEEE Transactions on Computers*, C-32(2):175–189, February 1983.

[12] R. Halstead and S. Ward. The MuNet: A Scalable Decentralized Architecture for Parallel Computation. In *Proceedings of the 7th Annual Symposium on Computer Architecture*, pages 139–145, May 1980.

[13] W. D. Hillis. *The Connection Machine*. The MIT Press, Cambridge, MA, 1985.

[14] Parviz Kermani and Leonard Kleinrock. Virtual Cut-Through: A New Computer Communication Switching Technique. *Computer Networks*, 3:267–286, October 1979.

[15] Leonard Kleinrock. *Queueing Systems.* John Wiley & Sons, 1975.

[16] Clyde P. Kruskal and Marc Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, December 1983.

[17] Clyde P. Kruskal, Marc Snir, and Alan Weiss. The Distribution of Waiting Times in Clocked Multistage Interconnection Networks. *IEEE Transactions on Computers*, 37(11):1337–1352, November 1988.

[18] James T. Kuehn and Burton J. Smith. The HORIZON Supercomputing System: Architecture and Software. In *Proceedings of Supercomputing '88*, November 1988.

[19] D. H. Lawrie. Access and Alignment of Data in an Array Processor. *IEEE Transactions on Computers*, C-24(12):1145–1155, December 1975.

[20] D. Lenoski, J. Laudon, K. Gharachorloo, A. Gupta, J. Hennessy, M. Horowitz, and M. Lam. *Design of the Stanford DASH Multiprocessor.* Computer Systems Laboratory TR 89-403, Stanford University, December 1989.

[21] A. Norton and G. F. Pfister. A Methodology for Predicting Multiprocessor Performance. In *Proceedings ICPP*, pages 772–781, August 1985.

[22] Janak H. Patel. Performance of Processor-Memory Interconnections for Multiprocessors. *IEEE Transactions on Computers*, C-30(10):771–780, October 1981.

[23] G. F. Pfister, W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, A. Norton, and J. Weiss. The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture. In *Proceedings ICPP*, pages 764–771, August 1985.

[24] Charles L. Seitz. Concurrent VLSI Architectures. *IEEE Transactions on Computers*, C-33(12):1247–1265, December 1984.

[25] Charles L. Seitz. The Cosmic Cube. *CACM*, 28(1):22–33, January 1985.

[26] Charles L. Seitz et al. The Architecture and Programming of the Ametek Series 2010 Multicomputer. In *Proceedings of the Third Conference on Hypercube Concurrent Computers and Aplications*, January 1988.

[27] Howard J. Siegel. *Interconnection Networks for Large-Scale Parallel Processing.* McGraw-Hill, 1990. Second Edition.

[28] H. Sullivan and T. R. Bashkow. A Large Scale, Homogeneous, Fully Distributed Parallel Machine. In *Proceedings of the 4th Annual Symposium on Computer Architecture*, pages 105–117, March 1977.

[29] C. D. Thompson. *A Complexity Theory for VLSI.* PhD thesis, Carnegie-Mellon University, Dept. of Computer Science, 1980.

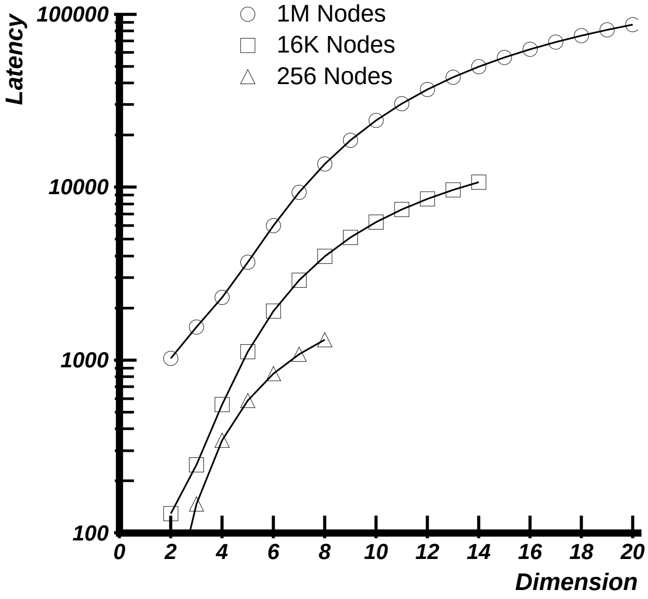


Figure 15: Latency for systems with 256, 16K, and 1M nodes assuming zero node delay ($s = 0$). Message lengths (L) are assumed to be 160 bits. Bisection width is normalized to a binary n-cube with unit-width channels.

A Keeping Bisection Width Fixed and Ignoring Node Delays

The analysis in this section keeps bisection width fixed, and assumes node delay is zero, as in [7]. Graphs are presented here for comparison with the case when switch delays are significant. With bisection width normalized to that of a binary n-cube with $W = 1$, the latency is given by

$$T_b = T^w(2)N^{\frac{1}{2}-\frac{1}{n}}\left(n\frac{N^{\frac{1}{n}}-1}{2} + \frac{2L}{N^{1/n}}\right)$$

Figures 15 and 16 compare latencies for various system sizes and various message lengths respectively, when switch delays are insignificant. It is clear that ignoring node delays favors low-dimensional networks.

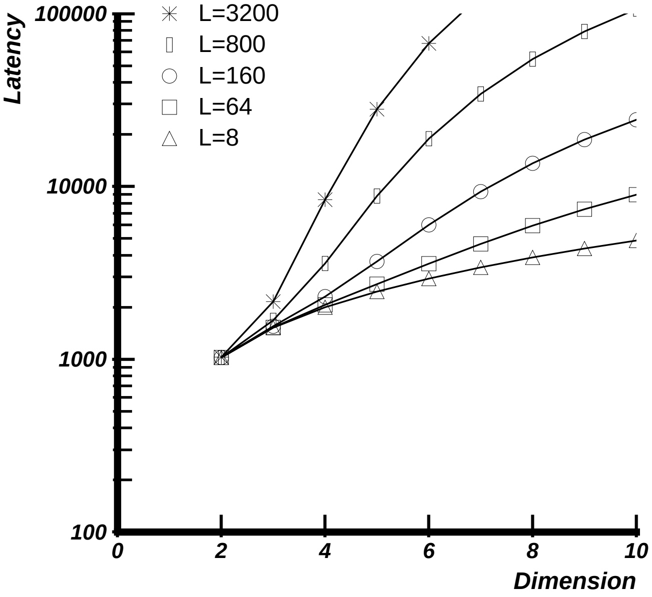


Figure 16: Latency for $1M$ node systems with message lengths L ranging from 8 through 3200 bits. Bisection width is normalized to a binary n-cube with unit-width channels.