A Family of Interconnection Networks for Non-Uniform Traffic

David M. Koppelman*
Louisiana State University
Baton Rouge, LA 70803

Abstract

New networks, called *GLO networks,* are constructed by adding *bus-like links* to omega networks, providing additional capacity between cells on momentarily busy paths. Equivalent pin-count GLO and omega networks offered uniform and nonuniform traffic were simulated. GLO networks exhibited lower latency for nonuniform traffic and light to moderate uniform traffic.

I. Introduction

A family of networks topologically similar to omega networks [8,10] but using links which operate as buses is described and analyzed; these networks are called generalized-link omega (GLO) networks. Some of the links in these networks operate as a bus, that is, they have a small number of inputs and outputs; devices at the inputs must contend for link use. Because the number of inputs and outputs is normally small, delays due to arbitration and fanout are minimal. The GLO networks are similar to omega networks in that if two cells (the network component containing a crossbar switch, queues, control hardware, etc.) are connected by a link in an omega network the corresponding cells in a GLO network will be connected by two links. The two links are called *simple links* and *bus-like links* (BLL), respectively. (Cells not connected with a link in an omega network are not connected in a GLO network.) The BLL's are intended to provide additional capacity between pairs of cells on a momentarily busy path. Any traffic pattern which distributes messages unevenly and unpredictably in an omega network might be better handled on a GLO network. Based on simulations this was the case for synthetic traffic patterns having uniform and slowly changing favorite destinations; the GLO networks had higher throughput and lower latency.

A GLO network achieves improved performance by providing effectively wider (more bits transferred in a single clock cycle) links between cells than a conventional omega network using the same number of connections (pins) per cell. (Although omega networks are being used as a basis for comparison, the results and discussion can apply to other banyan networks.) Since current implementation technology is pin-limited, pin count is a good measure on which to compare networks. With wider links for the same number of pins, GLO networks widen an important bottleneck encountered in omega networks.

A. Background

Omega and related networks have long been candidates for use in parallel computers (and for other applications) [2,9,10] to send data from a set of $N = m^s$ *inputs* to a set of $N = m^s$ *outputs* (where m and s are integers greater than one). These networks consists of s stages, each having m^{s-1} cells. The cells, all essentially identical, have m inputs and m outputs (called an $m \times m$ cell, m is the *degree* of the cell); they receive data in the form of *packets* through their inputs and emit the packets through their outputs. The cells in adjacent stages are connected by a set of N links, the link pattern determines the type of network (*e.g.,* omega, butterfly). A network constructed in this way with the link patterns such that there is exactly one path between every network input/output pair is called a *banyan network.* Of the various multistage networks, it is these which are most often considered for packet switching applications, including parallel computation.

* *IEEE Transactions on Parallel and Distributed Systems*, vol. 7, no. 5, pp. 486-492, May 1996.

Most previous research aimed at improving the performance of omega networks had been aimed at making more efficient use of the links between cells. The research was motivated by the small normalized throughput of a rudimentary omega network, about 25% of the network's capacity [3] (for the case of an 8-stage omega network using 2×2 single-buffered cells and offered traffic with a Bernoulli arrival process with uniformly distributed destinations). The normalized throughput is the fraction of link capacity being used. Were the normalized throughput 100%, which could be achieved for certain traffic patterns [8,9], every link would be transferring a packet at every clock cycle.

The reason for the low throughput is that messages are blocked in their trip through the network, in some cases leaving a link momentarily idle. A message can be blocked by another message in the same cell; a blocked message can block a message in a previous stage. The net result is that the speed of message movement drops closer to the inputs, limiting throughput [3,4,13].

Investigators increased throughput by providing and refining packet buffering within a cell. The earliest improvement was the simple provision of queues, raising throughput to about 75% [2,13]. More elaborate techniques employ multiple queues at each cell input (or output) which raises throughput to near 100% [1,11]. (These results are not directly comparable because of differences in message length and timing.) Another method to increase throughput is increasing the cell degree [6], thus reducing the number of stages. This technique is limited by the number of pins available using current fabrication technology and the area of the crossbar needed for switching within the cell.

B. The Link Bottleneck

In the techniques described above the links' capacity remains fixed. Improved packet buffering increases link usage, but this benefit is only realized when traffic is heavy enough for blocking to become a problem and does not address the link-width bottleneck. In contrast, the benefit of wider links in GLO networks is realized at light traffic levels and for non-uniform traffic. In many parallel computer applications minimum latency is of prime importance so that networks used for such applications will be lightly loaded. In such cases GLO networks would show improved performance. Further, the traffic in parallel computers would rarely have uniform destination distributions; an omega network would have to be over-built to handle such traffic.

The remainder of the paper is organized as follows. In the next section the basic structure of the GLO network is described. In Section 3 a cost analyses is presented; in Section 4 the simulation study and results are described. Conclusions appear in Section 5.

II. The GLO Network

The family of GLO networks has a topology similar to that of an omega network [8], the difference being the links between stages. (As used here, an omega network can be made of cells having more than two inputs and outputs.) The GLO-network topologies can be specified with three parameters: number of stages, cell size, and link patterns (between each stage). First, the system for describing link patterns will be described, followed by the networks to be studied.

A stage in an m^s-input omega network consists of m^{s-1} identical $m \times m$ cells; each cell (except those in the last stage) is connected to m cells in the following stage, one link for each connected pair. In GLO networks there are two sets of links between each stage, simple and bus-like links. Each set of links connects a cell to the same m cells as in the corresponding omega network.

The cells in adjacent stages can be divided into *groups* based on the links that would connect the cells in an omega network. Two cells are in the same group if, as in a graph, a path can be found from one cell to the other taking the cells in the adjacent stages as vertices and the links

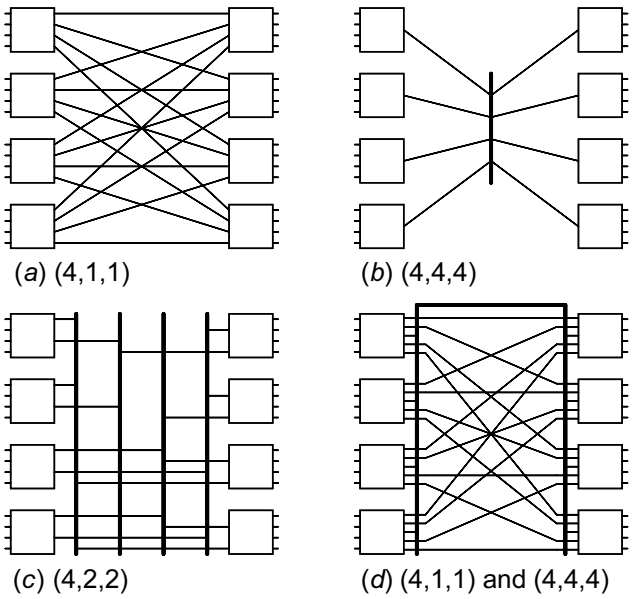

Figure 1. Some Link Patterns.

as edges. It can be shown that the number of cells in a group is $2m$. Every cell in one stage is connected to every cell in its group in the other stage.

The simplest set of links to connect cells in a group is a single bus connecting all cells in the group. Each cell in a network using this link pattern needs only one input and output. At the other end of the spectrum is the link set used in an omega network: a link for each pair of cells in adjacent stages and in the same group. Between these two extremes are a range of other possible link sets. See the end of Section 2 for an example network using these link sets. The family of possible link sets, along with other definitions, are described more precisely below. The definitions stress topology; functionality issues will be discussed further below.

A. Topology

Definition 1: A *link set* is a set of links. Given m distinct cells (called *input cells*) and m distinct cells (called *output cells*), an (m, a, b) link set is the smallest possible set of links, each link having a inputs and b outputs, such that for all input/output cell pairs there is a link in the set connecting the pair.

Lemma: Given m distinct cells (called input cells) and m distinct cells (called output cells), and given integers a and b such that $m \equiv 0 \bmod a$ and $m \equiv 0 \bmod b$ there exists an (m, a, b) link set containing $m^2/(ab)$ links such that each input cell is connected to m/b links and each output cell is connected to m/a links.

Proof: Evenly partition the set of output cells into m/b subsets. Each link has all of its outputs associated with all of the cells in one of the subsets. An input cell connects to exactly one link associated with each subset, for a total of m/b connections. Each subset will be connected to all the input cells, with a per link, so that m/a links per subset are needed.

For example, in Figure 1 *(a)*, cells are connected with a $(4, 1, 1)$ link set. This is the same link pattern used in an omega network built of 4×4 cells. (When an omega network is drawn, the cells in a stage are not necessarily arranged in groups.) In *(b)* cells are connected with a $(4, 4, 4)$ link set, which is a single bus, and in *(c)* cells are connected with a $(4, 2, 2)$ link set. In each case full connectivity is maintained. Assuming all links have the same width, the $(4, 1, 1)$ link set can carry the most data but using the most pins per cell; the $(4, 4, 4)$ link set can carry the least data although using the fewest pins per cell.

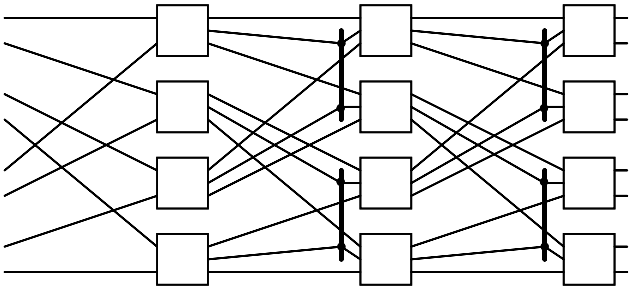

Figure 2. A $(3, 2, 2, 2)$ GLO Network.

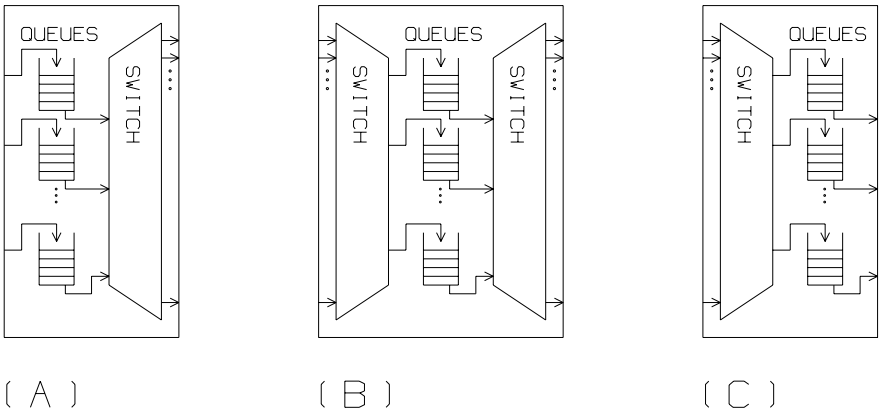

Figure 3. Possible Cell Configurations.

The examples cited so far have a single link set between cells. In Figure 1 *(d)* two link sets are used, $(4, 1, 1)$ and $(4, 4, 4)$. In a network using these link patterns the $(4, 1, 1)$ links might handle normal traffic, while the $(4, 4, 4)$ link might be used between a pair of cells with an unusually large amount of traffic.

Two link sets per group are used to connect the stages of GLO networks. The GLO networks' topology differs from an omega network in the link sets and the number of inputs and outputs per cell. The GLO networks will be defined more precisely below. The notation $\mathbf{Z}_x$ will be used to denote the set of integers from 0 to $x - 1$.

Definition 2: The topology of a GLO network can be specified by the four tuple (s, m, a, b), where s is a positive integer, m is a positive integer, and a and b are integers which divide m. This will be called the *GLO network description.* Let $N = m^s$. Such a GLO network consists of N terminals called network inputs and N terminals called network outputs; both sets of terminals are consecutively numbered starting from zero. There are s stages numbered from 0 to $s - 1$; each stage consists of m^{s-1} cells, numbered consecutively from zero. The cell inputs in stage zero are connected to network inputs by a simple link such that input i is connected to cell $i \bmod (m^{s-1})$. The simple links between stages form a shuffle connection. A shuffle connection is defined in terms of the m, k shuffle function, $\sigma_{m,k} \mid \mathbf{Z}_{mk} \rightarrow \mathbf{Z}_{mk}$, which is given by $\sigma_{m,k}(c) = (cm + \frac{c}{k}) \bmod mk$. For all $j \in \mathbf{Z}_{s-1}$, $i \in \mathbf{Z}_m$, and $c \in \mathbf{Z}_{m^{s-1}}$, cell c in stage j is connected to cell $\lfloor \sigma_{m,m^{s-1}}(cm+i)/m \rfloor$ in stage $j + 1$ by a simple link. Cells within a group are also connected using bus-like link sets (m, a, b). The cells in stage $s - 1$ each have m outputs; the outputs of cell c for $c \in \mathbf{Z}_{m^{s-1}}$ are connected by a simple link to network outputs $cm + i$ for all $i \in \mathbf{Z}_m$.

A GLO network can take many different forms. The GLO network $(4, 4, 4, 4)$ is like a 256-input omega network, except that it consists of 5×5 cells in the interior stages with cell groups connected as in Figure 1 *(d)*. The GLO network $(3, 2, 2, 2)$ consists of 8 inputs, four 2×3 cells in the first stage, four 3×3 cells in the second stage, and four 3×2 cells in the last stage. See Figure 2.

B. Cells

The additional links in GLO networks have two major implications for cell design. First, the switches within a cell which connect cell inputs and outputs to queues must be more elaborate. Second, the treatment of data entering the BLL's must be decided. The data can be treated either as an extension of one of the simple links, referred to as *split-channel BLL's;* or the data can be part of a separate data path, referred to as *independent-channel BLL's.* For split-channel BLL's the data sent in one clock cycle is split into two packets (one for the simple link and one for the BLL) and travel down simple links and BLL's simultaneously, the parts are reunited at a queue in the destination cell. For independent-channel BLL's data from a message would use either a BLL or a simple link between stages.

Three possible configurations of queues and switches are shown in Figure 3. In *(A)* the number of queues is equal to the number of inputs; the queues are placed at the inputs, and switches direct the packets to the proper outputs. In *(B)* the number of queues is different (usually less) than the number of inputs or outputs. A switch connects the cell's inputs to the queues, a second switch connects the queues to the cell's outputs. Finally, in *(C)* the queues are at the outputs.

Configurations *(A)* and *(C)* are more amenable to independent-channel BLL's. Configuration *(B)* in which m queues are used is more amenable to split-channel BLL's.

III. Analysis

A cost analysis will be described in this section. The cost analysis quantifies cost based upon a low-level hardware model; the goal is to demonstrate that GLO-network cells are comparable in cost to omega-network cells.

A. Cost Model

For a performance comparison between a GLO and omega network to be valid the networks compared must have equal cost. To that end a cost model will be described and applied to GLO networks using cell types B and C; the model partially applies to cell type A. The cost model for GLO and omega networks will be based on chip area and number of pins. Each chip will contain one cell. The number of pins needed for both data and control will be counted. Chip area will be based on the area taken up by the switch crosspoints. Expressions for the cost of both networks will be given in terms of network parameters (such as number of inputs and switch degree) and cost of crosspoints. The number of pins in an omega-network cell is given by

$$P(\Omega) = 2m(w+1), \tag{1}$$

where the factor of 2 accounts for both inputs and outputs, w is the physical width of the datapath (in bits), 1 accounts for a data-valid line, and Ω refers to an omega network using $m \times m$ cells with w bits per input. The number of pins in a GLO-network cell of type A, B, or C is given by

$$P(\mathrm{GLO}) = 2m(w_g+1) + \left(\frac{m}{a} + \frac{m}{b}\right)(w_g+3), \tag{2}$$

where w_g is the physical link width of the simple links and BLL's and GLO refers to the (s, m, a, b) GLO network described above. The first term accounts for those links which are also present in an omega network (but with width w_g) and the second term accounts for the bus-like links. Note that each bus-like link is assumed to use three pins for arbitration. (An exact arbitration mechanism is not specified. Arbitration might be done by propagating or blocking a link-grant token, using two pins. The third pin would be a data-valid signal.)

The cost of a switch will be estimated by counting crosspoints, one-bit switches. Each crosspoint will have cost C_{xp}. A w-bit wide, $x \times y$ switch then has cost $C_{\mathrm{xp}}\, xyw$. In cell types A and C

there is a connection from each of $m + \frac{m}{a}$ inputs to every one of the $m + \frac{m}{b}$ outputs. The cost of the switches in omega and GLO networks are

$$C_{\mathrm{XB}}(\Omega) = C_{\mathrm{xp}}\, m^2 w. \qquad \text{and} \qquad C_{\mathrm{XB}}(\mathrm{GLO}(C)) = C_{\mathrm{xp}} \left(m + \frac{m}{a}\right) \left(m + \frac{m}{b}\right) w_g.$$

For GLO networks of topology (s, m, a, b) using split channels, type-B cells, and m queues, the switches need not be complete crossbars. In fact, for the number of queues considered an output switch is not needed. Instead, each queue output connects to one simple-link output. Each bus-like link connects to b queue outputs. Note that each queue will be connected to one simple and one bus-like link. A queue can send data onto both links simultaneously. Queue inputs are similarly divided. One input is connected to an $m \times m$ crossbar, the inputs of this crossbar are connected to the simple-link cell inputs. The other queue input is connected to an $\frac{m}{a} \times m$ crossbar, the inputs of this crossbar are connected to the BLL inputs. The cost is given by

$$C_{\mathrm{XP}}(\mathrm{GLO}(B)) = C_{\mathrm{xp}} \left(m + \frac{m}{a}\right) m w_g$$

IV. Simulation Study

A simulation study was carried out to compare the performance of GLO and omega networks and to determine the effectiveness of various configurations on various traffic models. In the comparison study, omega and GLO networks using the same number of pins were simulated. Two traffic models were used: in one all message destinations are equally likely, in the other an input port has a stack of currently favorite destinations. The best GLO-network configuration was identified and the benefit quantified.

A. Methodology

The simulation was performed at the packet-transfer level; the simulator can simulate GLO and omega networks. The offered traffic consists of messages; the size of the messages (specified in bits) can be either constant or geometrically distributed. Links can transfer a fixed number of bits in a clock cycle, the width of the link. The simulated network uses cells of type B with queues at the outputs; because packets of varying size can enter and leave a queue, a queue's memory is not organized as slots but rather as a pool. The queue size was 600 bits for all simulations performed. The input of each queue is connected to the output of a switch; in a single cycle a queue can receive data that is part of only one message. (That is, all or part of two or more messages cannot enter a queue in one cycle.) The simulator allows virtual-channel flow control and complete connection of inputs to queues [1], that is, there can be multiple queues per port where each queue in a port can receive a packet during a cycle.

A packet is transferred into a queue at time t if the corresponding queue has sufficient space free at time $t - 1$. (Time is the number of clock cycles completed since the start of the simulation.) The network uses virtual cut-through routing [10], so that the head packet of a message can leave a cell before the entire message is in the cell. When a message's head packet reaches the head of a queue it requests use of a switch output. If the requested output is not blocked the request is granted for the next clock cycle. The arbitration process is repeated every cycle, preventing a blocked queue from delaying any other queues sharing a port.

BLL arbitration is accomplished by examining each queue that could connect to a BLL. The queue with the most items and which was not blocked at arbitration time is chosen. The simulator uses network state at time $t - t_{\mathrm{bll-lat}}$ to determine bus usage at time t, where $t_{\mathrm{bll-lat}}$ is the time taken for arbitration. For the networks discussed below arbitration time is set to $t_{\mathrm{bll-lat}} = 2$, twice as long as the time to resolve a switch connection request.

Once a connection is established packets are transferred, one per clock cycle, as long as the destination queue has sufficient room during the previous clock cycle. The estimate made is conservative: there must be enough room to receive data as if a BLL were assigned to the queue. In the last stage of the network packets are transferred to the network outputs; these are never blocked.

In omega networks all cells are nearly identical. The number of inputs and outputs of cells in a GLO-network varies. If a GLO network were implemented with one cell per chip, then the cells in the input and output stages would use fewer pins than in a corresponding omega network (if link widths were uniform). Rather than use chips with fewer pins, the links could be widened at the network inputs and outputs, thus using the same number of pins on all chips.

The widths of the simulated network input, output, internal simple, and internal bus-like links can all be independently set. These were adjusted to simulate equivalent-pin-count networks and to evaluate networks with higher capacity links at inputs and outputs.

Messages arriving at the network inputs are placed in queues; the number of queues per input is the same as the number of queues per port. (In an n stage network there are $n+1$ stages of queues.) The size of the queues is the same as other queues in the network, with an important exception. If an arriving message is blocked because of a full input queue, it is placed in a second infinite queue (actually, a counter). This was done to prevent the simulator from "filtering" the offered traffic. (In an actual system a processor whose messages are not delivered might stall. When the messages are delivered and responded to it would resume, with the pattern of destinations probably the same as if it hadn't been forced to stall [5].)

Of the data the simulator collected the most important are latency and throughput. The message latency for a simulation is the average of the number of cycles between the time a message is placed in a network input (finite) queue and the time its last packet is removed from the last stage. The throughput of a simulation is $b_{\text{total}}/(Nt_{\text{sim}}w_i)$, where b_{total} is the total number of bits received at the network outputs, t_{sim} is the number of cycles the simulation was run for, and w_i is the width of the links connecting the inputs to the first stage. This definition of throughput is non-standard because it is based on first-stage link width; this was done because of the varying link widths of the networks under study. For networks with uniform link widths and messages lengths which are an integral multiple of link width this definition is equivalent to the standard definition of normalized throughput: average number of packets entering a network input per cycle [3].

Let the average message length of traffic offered to a network be denoted $\overline{L}$. Then the *traffic intensity* is defined to be $\rho = \lambda\overline{L}/w_i$. As with throughput this definition takes into account the varying link widths.

The traffic model used for the simulation has geometrically distributed message lengths and interarrival times, and both uniform and non-uniform destination distributions. The message length and interarrival time distribution used is commonly employed to simulate may types of networks [2,7,10]. The non-uniform destination distribution, to be described in detail below, was chosen to capture the message traffic behavior which the GLO network is designed to accommodate.

The non-uniform traffic model is similar to one proposed by Thiébaut for cache simulation [12]. The process that generates the destinations consists of a stack of destinations and a geometrically distributed random variable which indexes the stack. (In [12] a hypergeometric random variable is used and the stack contents are memory addresses instead of destinations.) Each entry in the stack is initialized with a randomly chosen destination (using a uniform distribution). Destinations are generated by successively sampling the random variable. A destination is the stack entry pointed to by the random variable. After each sample the stack is modified by moving the indexed stack entry to the top of stack. For example, if the stack initially contained destinations [7,6,3,4,1] (with the leftmost element being at the top of stack, having index 0), and the random variable took on values [0,1,0,3,3] then the trace would consist of destinations [7,6,6,4,3] and the stack would contain destinations [3,4,6,7,1] after the last sample.

This model is useful for capturing varying amounts of temporal locality while also allowing for slow change in destination frequency over time. A destination near the top of stack at some point in time will be appear frequently near that time, but that same destination may later sink to a lower position and so appear less frequently.

The simulator uses the stack model to generate non-uniform destinations. An important property of the destination distribution is its uniformity. Uniformity is determined by the parameter $p \in [0, 1]$. The probability of the i'th stack entry being chosen at any cycle is $p(1-p)^i$. Destinations generated with smaller p are more uniform, those generated with larger p are limited to fewer favorite destinations. Each input is associated with its own stack of destinations, initialized to random values. The value of p is identical for all inputs.

B. Experiments

In the simulation study equivalent-pin-count GLO and omega networks of a variety of cell configurations were compared under a variety of conditions. All the omega networks used 2×2 cells, and all the GLO networks were of topology $(8, 2, 2, 2)$. Of these, two configurations will be described in detail here. One configuration, to be referred to as dual channel here, consists of networks using virtual-channel flow control (with two channels per port) and nonblocking switches. The other configuration, to be referred to as single channel, uses a single queue per port.

For each simulation the number of pins used, $P_{\max}$, will be 168 or 188 (depending on the cell, see below), the number of pins in a medium-sized chip package. Values of w and w_g were computed so that all available pins would be used (based on (1,2)).

Forcing the number of pins on the two networks to be equal does not make the chip area equal; in fact the GLO network has slightly larger area. The difference in area is not that great; less than a factor of two for each network examined. Furthermore, whereas the chip area available grows with the square of the number of pins, it can be shown that the complexity of the chip area used by the GLO network is only a logarithmic factor greater than the square of the number of pins. Therefore the area relationship between the two networks will scale.

In both configurations the network had 8 stages of 2×2 cells and 600-bit queues. Three link widths were used: both networks used 52-bit links to connect cells to network inputs and outputs. The omega networks used 41-bit links to interconnect cells, while the GLO networks used 26-bit simple links and 26-bit BLL's. These widths were chosen so that corresponding cells in the GLO and omega networks would have the same number of pins, assuming that simple links each have one pin for control information and that BLL's each have three pins for control information. (The GLO-network link widths were rounded down from $26\frac{1}{3}$.)

The link widths for the cells not connected to inputs and outputs were based on 168-pin chips. Because they were bottlenecks, the links at inputs and outputs were made wider, so that the cells to which they would connect would require additional pins, 188 v. 168 pins. Networks using wider input and output links will be referred to as *flared* networks. Flaring improves the performance of both GLO and omega networks, however performance improvement is greater in GLO networks.

The configurations differ on the principle source of blocking. In the single-channel configuration the switches will contribute to blocking since data entering two inputs cannot be directed towards the same port at once. In the dual-channel configuration data entering two inputs can be directed towards the same port, with each directed into a different queue. Two queues cannot use a link at once, so the links are the principle source of blocking. Note that link capacity is identical in both configurations.

The simulations were performed using both uniform and non-uniform destination distributions over a range of arrival rates. Message lengths were geometrically distributed with a mean of $\overline{L} = 205$ bits. Arrival rates varied from $\lambda = 0.005$ messages per input per cycle to saturating (always a message waiting at every input). Non-uniform traffic was generated using the stack

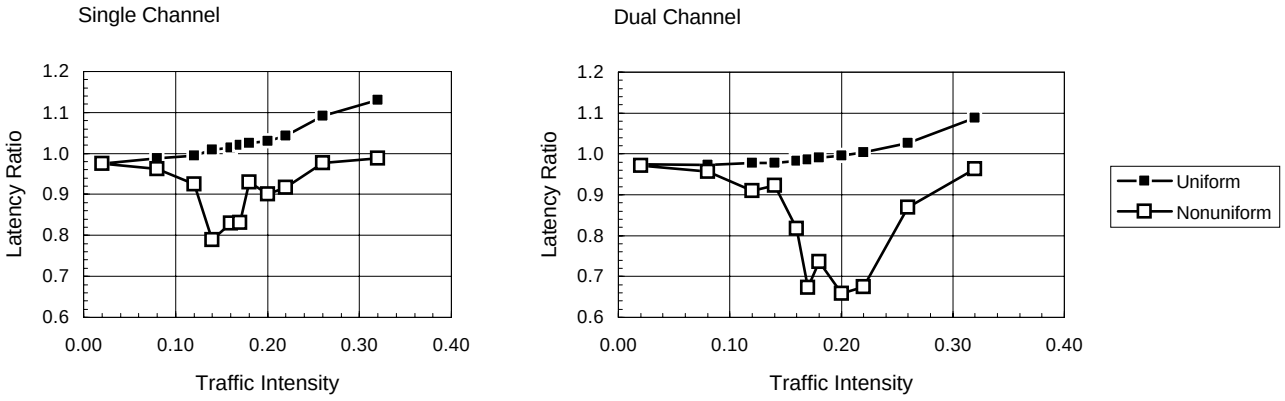

Figure 4. Traffic Ratios.

model described above, with $p = .9$ and using a three-entry stack (in case of stack overflow a randomly chosen destination is used).

Each simulation was run eight times for 10 000 cycles. The simulator collected, among other performance data, the throughput, latency, and average amount of data transferred over each link. The data presented below is based on a mean of the eight runs. For heavy non-uniform traffic conditions the 95% confidence interval for some data was large, over 10% of the sample mean, for other conditions the confidence interval was much smaller. Confidence intervals are noted in the table.

C. Results

The GLO network performed better than the omega network for all simulations using light traffic. The GLO network outperformed the omega network for all simulations using non-uniform traffic as well. The omega network outperformed the GLO network only for moderate to heavy uniform traffic. This is illustrated in Figure 4 where the ratio of latencies is plotted against traffic intensity for both configurations and for uniform and non-uniform traffic. Note that the difference between the two networks is small when offered uniform traffic, with the omega network in the single-channel configuration performing better (ratio greater than one). For non-uniform traffic GLO networks have significantly lower latencies, 75% or less for moderate traffic. The ratios are large at lower traffic intensities; for higher intensities the ratios are smaller; the minimum occurs at the highest traffic intensity that the network can comfortably handle. At high traffic intensities the ratio increases because of the increase in fraction of time both of a cell's ports are needed. The advantage of a GLO network over an omega network is greater for the dual-channel configuration.

Table 1

Traffic Type	Queue Type	Network Type	Light Traffic		Moderate Traffic		Saturating Traffic	
			ρ	Latency	ρ	Latency	Throughput	Latency
Non-uniform	Dual-Channel	GLO	.02	13.4 ± 0.1	.20	49.3 ± 5.0	.246 ± .0058	289 ± 7.1
		Omega	.02	13.8 ± 0.1	.20	74.8 ± 8.9	.227 ± .0048	303 ± 5.8
	Single-Channel	GLO	.02	13.5 ± 0.1	.14	31.1 ± 2.2	.190 ± .0028	185 ± 4.1
		Omega	.02	13.8 ± 0.1	.14	39.4 ± 3.7	.188 ± .0037	185 ± 3.7
Uniform	Dual-Channel	GLO	.02	13.3 ± 0.1	.20	21.6 ± 0.1	.422 ± .0006	188 ± 0.3
		Omega	.02	13.7 ± 0.0	.20	21.8 ± 0.1	.457 ± .0013	161 ± 0.4
	Single-Channel	GLO	.02	13.4 ± 0.0	.14	20.1 ± 0.1	.290 ± .0009	123 ± 0.3
		Omega	.02	13.7 ± 0.0	.14	19.9 ± 0.1	.300 ± .0011	114 ± 0.3

The ratios plotted in Figure 4 show relative performance; actual latency and throughput for light, moderate, and saturating traffic are tabulated in Table 1. The data for moderate traffic is taken at the highest traffic intensity (for which simulations were run, no attempt was made to interpolate) for which throughput is 95% or more of offered traffic. At higher traffic intensities the number of items in the input queues is large. The data for saturating traffic intensity was based on simulations in which there would always be message waiting at every input.

As can be seen in Table 1 the latencies are almost identical for the two networks under light traffic. The latencies are dominated by the delay due to the nine stages of links, present in both networks. The latency in the GLO networks is lower due to the fewer transfers necessary to move a message across a link. More important is the lower latencies at moderate traffic levels. The GLO-network latency is significantly lower for non-uniform traffic. Omega networks have lower latency than GLO networks at saturating uniform traffic. GLO networks have lower latency when offered non-uniform traffic.

The throughput of the GLO networks is higher for the dual-channel case but lower for the single-channel case. This suggests that switch blocking, which occurs more in the single-channel case, reduces the effectiveness of GLO networks.

From these simulations it can be concluded that for non-uniform traffic GLO networks have a significant advantage in latency, particularly if the networks use virtual channels. For uniform traffic conditions an omega network is slightly better than an equivalent-pin-count GLO network for moderate and heavy traffic conditions.

V. Conclusions

A family of networks, called the GLO networks, was described. These networks are characterized by their use of bus-like links. The BLL's provide complete connectivity at low cost, but can block certain paths. The BLL's are intended to provide a wide datapath for transient or long-lasting non-uniformities in traffic. A simulation study was undertaken to evaluate the network. The networks were simulated to determine their effectiveness under uniform and non-uniform traffic conditions for a variety of configurations. In comparison to omega networks using an equivalent number of pins and offered light traffic these networks have lower latency and comparable throughput. Further, GLO networks have better performance for non-uniform traffic of light to saturating arrival rates.

The networks' performance under non-uniform traffic is of value because the parallel computers for which these networks are intended generate such traffic. Minimum latency is of prime importance in such systems.

The GLO networks are an improvement over existing networks, especially for use in parallel computation where uniform traffic is uncommon. The research described here demonstrates that networks can be built for non-uniform traffic without simply scaling clock speed or link width.

VI. Bibliography

[1] W. J. Dally, "Virtual-channel flow control," *IEEE Transactions on Parallel and Distributed Systems,* vol. 3, no. 2, pp. 194–205, March 1992.

[2] D. M. Dias and J. R. Jump, "Packet switching interconnection networks for modular systems," *IEEE Computer,* vol. 14, no. 12, pp. 43–54, December 1981.

[3] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Transactions on Computers,* vol. 30, no. 4, pp. 273–282, April 1981.

[4] Y. C. Jenq, "Performance analysis of a packet switch based on single-buffered banyan network," *IEEE Journal on Selected Areas in Communications,* vol. 1, no. 6, pp. 1014–1021, June 1983.

[5] E. J. Koldinger, S. J. Eggers, and H. M. Levy, "On the validity of trace-driven simulation for multiprocessors," *ACM Computer Architecture News,* vol. 19, no. 3, pp. 244–250, May 1991.

[6] C. P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Transactions on Computers,* vol. 32, no. 12, pp. 1091–1117, December 1983.

[7] C. P. Kruskal, M. Snir, and A. Weiss, "The distribution of waiting times in clocked multistage interconnection networks," *IEEE Transactions on Computers,* vol. 37, no. 11, pp. 1337–1352, November 1988.

[8] D. H. Lawrie, "Access and alignment in an array processor," *IEEE Transactions on Computers,* vol. 24, no. 12, pp. 1145–1155, December 1975.

[9] F. T. Leighton, "Introduction to parallel algorithms and architectures: arrays * trees * hypercubes," Palo Alto: Morgan Kaufmann, 1992.

[10] H. J. Siegel, W. G. Nation, C. P. Kruskal, and L. M. Napolitano, jr., "Using the multistage cube topology in parallel supercomputers," *Proceedings of the IEEE,* vol. 77, no. 12, pp. 1932–1953, 1989.

[11] Y. Tamir and G. L. Frazier, "Dynamically-allocated multi-queue buffers for VLSI communication switches," *IEEE Transactions on Computers,* vol. 41, no. 6, pp. 725–737, June 1992.

[12] D. Thiebaut, J. L. Wolf, and H. S. Stone, "Synthetic traces for trace-driven simulation of cache memories," *IEEE Transactions on Computers,* vol. 41, no. 4, pp. 388–410, April 1992.

[13] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance analysis of multibuffered packet-switching networks in multiprocessor systems," *IEEE Transactions on Computers,* vol. 39, no. 3, pp. 319–327, March 1990.