SIAM J. COMPUT.
Vol. 11, No. 2, May 1982

0097-5397/82/1102-0012 $01.00/0

A SCHEME FOR FAST PARALLEL COMMUNICATION*

L. G. VALIANT†

Abstract. Consider $N = 2^n$ nodes connected by wires to make an n-dimensional binary cube. Suppose that initially the nodes contain one packet each addressed to distinct nodes of the cube. We show that there is a distributed randomized algorithm that can route every packet to its destination without two packets passing down the same wire at any one time, and finishes within time $O(\log N)$ with overwhelming probability for all such routing requests. Each packet carries with it $O(\log N)$ bits of bookkeeping information. No other communication among the nodes takes place.

The algorithm offers the only scheme known for realizing arbitrary permutations in a sparse N node network in $O(\log N)$ time and has evident applications in the design of general purpose parallel computers.

Key words. network routing, Monte Carlo algorithm, randomization, parallel computers

1. Introduction. We propose a solution to a fundamental communication problem. Suppose that N devices connected together by a sparse network of wires wish to communicate amongst themselves simultaneously. Suppose also that the communication pattern is unpredictable and rapidly changing as may be required, for example, when the devices are computers cooperating in executing a parallel algorithm. The problem is to specify a network topology and a routing algorithm that can implement arbitrary such communication requests efficiently.

In particular we consider the paradigmatic communication requirement of a permutation. Each device is a node of the network and has a distinct name x from the set $\{0, 1, \cdots, N-1\}$. Initially node x contains a "packet" labelled by an address $a(x) \in \{0, 1, \cdots, N-1\}$ that is the destination node to which the packet is to be sent. If the N addresses are all distinct then the communication requirement is a permutation.

The contraint of having only a few wires from each node, and hence a sparse graph is dictated by physical limitations. It implies that it will take a long time to gather complete information about the permutation request at any one node, as this would have to be done largely sequentially. This strongly suggests that we need to look for a distributed routing strategy that does not require any node ever having more than fragmentary information about the permutation.

There are several quantitative criteria according to which such parallel communication schemes (PCSs) may be judged. The scheme that we propose provably achieves the following parameters:

1. *Speed.* Every permutation can be implemented in $O(\log N)$ steps. (By a step we mean the time taken to transmit a packet along a wire. Computations carried out locally at a node are not counted here).

2. *Sparsity.* Each node has $O(\log N)$ wires from it.

3. *Simplicity.* The bookkeeping information carried with each packet (e.g., its address) is small ($O(\log N)$ bits). Such information is never transmitted except when accompanying a packet. The local computations needed at each node are easy and efficiently parallelizable.

4. *Flexibility.* (i) Besides complete permutations it can also implement partial ones. (ii) No global synchronization is required.

* Received by the editors October 15, 1980 and in revised form March 24, 1981.
† Computer Science Department, Edinburgh University, Edinburgh, Scotland EH8 9YL.

There are essentially only two previously known constructions on which rival schemes could be based: Batcher's sorting networks [2] and the permutation networks of Benes [3], [10]. The former suffers from the disadvantages that it requires $\Omega(\log^2 N)$ steps and cannot directly do partial permutations. The problem with the latter is that the only fast parallel routing algorithms known require global information and take time $\Omega(\log^2 N)$ even on a parallel random access model of computation [6]. An advantage they do share over our scheme is that of greater simplicity in the local computations, but this appears to be of ever diminishing relevance for currently anticipated technologies.

In our scheme the network topology is simply the n-dimensional binary cube. We therefore consider values of N with $N = 2^n$ for some integer n. Between every pair of adjacent nodes of the cube we draw a pair of oppositely directed edges to represent communication in the two directions. This topology has been suggested frequently before. For a survey and bibliography see Siegel [8]. Previous schemes all require at least $\Omega(\log^2 N)$ steps for some inputs.

The claimed speed of our scheme needs one qualification. The routing algorithm makes random choices in the same sense as the famous primality tests of Strassen and Rabin [7], [9]. It is correct for all inputs (i.e., permutations) and, for some constant C will terminate in $C \cdot \log_2 N$ steps with overwhelming probability. A noteworthy feature of the algorithm is that its runtime distribution is provably *identical* for every input. The algorithm is therefore *testable* in the sense that its general behavior for a fixed N can be determined with great confidence by running it often enough on any one input (even the identity permutation!). Since the analytic techniques we use, or can envisage, yield relatively crude complexity bounds, making comparisons among refinements of the algorithm is probably possible only by experimentation. We emphasize that here experimental results can be given a rigorous interpretation, a circumstance that we have not met before in as strong a sense in the context of algorithms.

2. Outline of the algorithm. In describing the algorithm we identify each packet by its starting node. Denoting the set $\{0, \cdots, N-1\}$ by V, the name of each packet is therefore a number $s \in V$.

The algorithm consists of two phases run consecutively. Phase A sends each packet $s \in V$ to a randomly chosen node $u(s) \in V$. For each s every $u \in V$ has the same probability (i.e., $1/N$) of being chosen, and the choices for the different packets are independent of each other. The second phase then routes each packet s from $u(s)$ to its correct destination $t = a(s)$.

At each instant there is just one copy of each packet, and this is either (a) being *transmitted* along an edge, or (b) wating in a *queue* associated with such an edge, or (c) stored as *loose* at a node.

For simplicity of exposition the algorithm is described in synchronized fashion although this is inessential. In this form the algorithm alternates between being in a transmitting mode and a bookkeeping mode. In the former case the packet at the head of each queue is transmitted along the edge associated with it and stored as loose at the recipient node. In the bookkeeping mode each loose packet is assigned to the queue of one of the outgoing edges according to some random choice, unless it has nowhere further to go in the current phase. (For a description of this algorithm in less synchronized form see [11], [12].)

In Phase A each packet makes for itself a random ordering of the n dimensions. It considers each one in turn and according to the toss of a coin makes, or refrains from making, a *move* in that dimension from its current position. (By making a move

we mean here that we add it to the appropriate queue. Actual transmission may be delayed by the presence of other packets in the queue.) It is immediate that with this procedure for any fixed packet every node has the same probability of being its destination. What needs to be proved is that no packet will have to wait in queues for more than $O(n)$ steps.

Phase B is similar except now each packet considers the set of dimensions in which its current location differs from its final destination, and moves along one randomly chosen such dimension at each step. Correctness is again immediate. What needs to be proved is that under the assumption that the packets are initially at randomly chosen nodes (as guaranteed by Phase A) no packet will wait in queues for more than $O(n)$ steps.

Analysis similar to Lemma 1 in § 4 shows that in each phase the probability that Cn different routes visit any one node is bounded above by $\exp\{-Cn/4\}$. This gives a crude upper bound on the maximum number of packets that may reside at any one node at any one time.

3. The algorithm. The n-dimensional cube will be represented by the set $V = \{0, 1, \cdots, N-1\}$ of $N = 2^n$ vertices. For $i \in \{1, \cdots, n\}$ and $x \in V$, x^i denotes the ith most significant bit in the n-bit binary representation of x. Also $x/\!/i$ is the number obtained by changing the ith bit of the binary representation of x to its complement. The $n2^n$ edges of V are therefore the pairs

$$\{(x, x/\!/i) | x \in V, i \in \{1, \cdots, n\}\}.$$

For each such edge there is a "Queue(x, i)" that feeds it and resides at node x. At each node x there is a set "Loose$_x$" in addition to the n queues.

In the algorithm the subroutine call "Transmit x" means 'for each i transmit the packet at the head of Queue (x, i) to node $x/\!/i$ and add it to the set Loose$_{x/\!/i}$'.

"Pick $d \in D$" means 'assigning equal probability to each member of set D choose a random element of D and assign it to variable d'.

Each packet $v \in V$ is associated with a set $T \subseteq \{1, \cdots, n\}$. In Phase A it consists of the set of dimensions along which possible transmissions have not yet been considered. In Phase B it is the set of dimensions along which transmission still has to take place. Each of the phases is said to be *finished* when for every $v \in V$ T_v is empty.

The routing algorithm consists of calling Phase A followed by Phase B, with the constants F, G chosen large enough that both algorithms finish with overwhelming probability. Note that when Phase A is finished all the queues are empty, and Loose$_x$ contains the set of packets randomly assigned to node x. Hence when Phase B is entered all the packets are loose and the queues empty.

The algorithms are described in Algol-like notation. Parallel execution of a block with variable d ranging over set D is denoted by

For $d \in D$ **cobegin** $\cdots$ **coend**.

The reader can verify by inspection that both phases are correct: when the first one finishes the packets are at independently chosen random nodes, after the second one they are all at their final destinations.

We remark that neither correctness nor the subsequent analysis depends on the particular disciplines used for maintaining the queues or the loose sets. The only assumption is that they are sets in which elements can be added or taken away. A second remark is that, in practice, the innermost loop could by implemented by a special purpose chip exploiting parallelism rather than by a sequential computation.

Phase A

For $s \in V$ **cobegin** $\text{Loose}_s := \{s\}$.
$T_s := \{1, \cdots, n\}$.
coend
For $f := 1$ **step** 1 **until** F **do**
For $s \in V$ **cobegin if** $\text{Loose}_s \neq \emptyset$ **then for**
each $v \in \text{Loose}_s$ **with** $T_v \neq \emptyset$ **do**
begin Pick $i \in T_v$.
$T_v := T_v - \{i\}$.
Pick $\alpha \in \{0, 1\}$.
if $\alpha = 1$ **then**
begin add v to Queue(s, i).
$\text{Loose}_s := \text{Loose}_s - \{v\}$.
end.
end.
Transmit s.
coend

Phase B

For $x \in V$ **cobegin if** packet with address x is at node u
then $T_x := \{i | x^i \neq u^i\}$.
coend
For $g := 1$ **step** 1 **until** G **do**
For $u \in V$ **cobegin if** $\text{Loose}_u \neq \emptyset$ **then for**
each $v \in \text{Loose}_u$ **with** $T_v \neq \emptyset$ **do**
begin $\text{Loose}_u := \text{Loose}_u - \{v\}$.
Pick $i \in T_v$.
$T_v := T_v - \{i\}$.
Add v to Queue(u, i).
end
Transmit u.
coend.

4. Analysis of the algorithm. The aim of the analysis is to show that for a sufficiently large constant C the routing algorithm will finish within $2Cn$ steps with overwhelming probability.

THEOREM. *For any constant S there is a C such that for $F = G = Cn$ both phases of the routing algorithm finish with probability greater than* $1 - 2^{-Sn}$.

For the analysis we need some facts from probability theory. Suppose that we have N independent Bernoulli trials each with probability p. Then the probability $B(m, N, p)$ that at least m of the trials are successful is bounded above by the normal distribution in the following way [1]:

Fact 1. If $m = Np(1+\beta)$ where $\beta \in [0, 1]$ then $B(m, N, p) \leqq e^{-\beta^2 Np/2}$.

We shall be interested in independent trials with varying probabilities (i.e., Poisson trials). If we have N such trials with probabilities $p_1, \cdots, p_N$, such that $\Sigma p_i = Np$ then, as is well known, the variance in the number of successes is maximal when $p_1 = p_2 = \cdots = p_N = p$. The following theorem of Hoeffding [5] is a stronger version of this.

Fact 2. If T is the number of successes in N independent Poisson trials with probabilities $p_1, \cdots, p_N$, then if $\Sigma p_i = Np$ and $m \geqq Np + 1$ is an integer, then

$$\Pr(T \geqq m) \leqq B(m, N, p).$$

For combinatorial formulae we shall use the notation n_r to denote $n!/(n-r)!$, and $\binom{n}{r}$ to denote $n_r/r!$. From elementary considerations it is easy to verify the following:

Fact 3. For all n

$$\sum_{r=1}^{n} \frac{1}{\binom{n}{r}} \leqq \frac{5}{3}.$$

Fact 1 is an estimate of the tail of the binomial distribution near the mean. For our main theorem we need estimates further from the mean and for this we use the following bound.

Fact 4. If $n \geqq Np$ is an integer then

$$B(m, N, p) \leqq \left(\frac{Np}{m}\right)^m \cdot e^{m-Np}.$$

Proof. Chernoff's bound [4] is

$$B(m, N, p) \leqq \left(\frac{Np}{m}\right)^m \left(\frac{N-Np}{N-m}\right)^{N-m}.$$

Putting $x = (N-m)/(m-Np)$ and using $(1+x^{-1})^x < e$ gives the required bound. □

The analysis of the two phases A and B are rather similar and will be given in tandem. The basic notion is that of a *route*, denoted typically by $R = \{e_1, e_2, \cdots, e_h\}$ where each e_i is an edge (x_i, y_i). A route is any path in the cube in which no two edges traverse the same dimension, i.e., if for some k

$$x_i /\!\!/ k = y_i \quad \text{and} \quad x_j /\!\!/ k = y_j$$

then $i = j$. Thus routes are minimum distance (acyclic) paths between their end points.

For any fixed route R and node $s \in V$ the event that "in running Phase A at least one edge occurring in the route from s also occurs in R" will be denoted by "$|R \cap s\rightarrow| \geqq 1$". For any fixed route R and node $t \in V$ the event that "in running Phase B, with the packet destined for t initially at a randomly chosen node, at least one of the edges occurring in the route of that packet also occurs in R" will be denoted by "$|R \cap \rightarrow t| \geqq 1$". More generally, "$s\rightarrow$" denotes the route from s, "$\rightarrow t$" the route to t, and "$x \rightarrow y \rightarrow z$" a route from x via y to z. The intersection $Q \cap R$ of two routes Q and R is the set of common edges. "R through x" is the event that the route R goes through x.

The first lemma bounds the number of routes that intersect with any one route.

LEMMA 1. *For all* $R = \{e_1, \cdots, e_h\}$ *and* $C \geqq 1$

(A) $\Pr(|R \cap s\rightarrow| \geqq 1$ *for at least* Cn *values of* $s) \leqq \exp\{-Cn/4\}$.

(B) $\Pr(|R \cap \rightarrow t| \geqq 1$ *for at least* Cn *values of* $t) \leqq \exp\{-Cn/r\}$.

Proof. (A) Let

$$p_s = \Pr(|R \cap s\rightarrow| \geqq 1) \leqq \sum_{i=1}^{h} \Pr(e_i \in s\rightarrow) = \sum_{i=1}^{h} p_{si} \quad \text{say.}$$

Since the $n2^n$ edges in the cube have identical roles one can argue that, by symmetry,

$$\sum_{s \in V} p_{si} = \sum_{s \in V} p_{sj}$$

for any $i, j \in \{1, \cdots, h\}$. Hence

$$\sum_{s \in V} p_s \leqq h \sum_{s \in V} p_{s1}.$$

But $\sum_s p_{s1}$ is simply the fraction $1/(n2^n)$ of the expected total number of edges occurring in the 2^n routes. Since the expected number of edges on each route is $n/2$,

$$\sum_{s \in V} p_s \leqq h \sum_{s \in V} p_{s1} \leqq \frac{h}{2}.$$

We therefore have N independent Poisson trials with respective probabilities $p_0, p_1, \cdots, p_{N-1}$ that have sum $h/2$.
By Fact 2

$$\Pr(\text{at least } Cn \text{ successes}) \leqq B\left(Cn, N, \frac{h}{(2N)}\right) \leqq B\left(Cn, N, \frac{Cn}{2N}\right)$$

since $h \leqq n$ and $C \geqq 1$. Hence applying Fact 1 with $\beta = 1$

$$\Pr(\text{at least } Cn \text{ successes}) \leqq \exp\left(-\frac{Cn}{4}\right).$$

(B) Let $p_t = \Pr(|P \cap \to t| \geqq 1)$. Then by the same argument as used in (A) we get

$$\sum_{t \in V} p_t \leqq \frac{h}{2}$$

and hence the required result. □

We note that the packet routes in Phase B when viewed in reverse are identical to the packet routes in Phase A. The source nodes in Phase A, like the targets in Phase B, represent each node of the graph exactly once. The targets in Phase A, like the source nodes in Phase B, represent random mappings of the N packets to the N nodes. Since our proofs concern only the routes themselves, independent of timing considerations, the proofs for Phase B will be always essentially the same as for Phase A.

If we could assume that two paths never intersect more than once then Lemma 1 would suffice to prove the Theorem. Unfortunately this is not the case. The following rough argument shows that in Phase A with large probability at least one pair of routes will be identical for about $n/\log n$ consecutive edges: consider two routes from neighboring starting nodes. With probability $(2n)^{-1}$ the first packet goes to the starting node of the second at the first step, and with probability $((n-1)_r)^{-1}$ it then follows the second path for r steps (provided both make r steps, which is most likely if $r = n/\log n \ll n/2$). But $(2n_{r+1})^{-1}$ exceeds $2^{-(n-1)}$ if $r \lesssim n/\log n$, and there are 2^{n-1} such pairs of routes to consider. It is therefore likely that at least one such pair follow each other for $n/\log n$ steps.

We therefore need the following lemma to bound the probability of two routes having r edges in common.

LEMMA 2. *Let R and Q be routes of length j from node x to node y. Suppose that R is fixed and Q is randomly chosen from all such routes. Then for $K = \frac{5}{3}$*

$$\Pr(|R \cap Q| \geqq r) \leqq \frac{K^r}{j_r}.$$

Proof. The result is established by induction on r. It is clearly true for $r = 0$. Suppose it is true for $r = 0$. Suppose it is true for $r-1$. Let $R = e_1 e_2 \cdots e_j$. Then $\Pr(|R \cap Q| \geqq r)$ is less than

$$\sum_{m=1}^{j-r+1} \Pr(|e_{m+1} \cdots e_j \cap Q| \geqq r-1 \,|\, e_m \in Q) \cdot \Pr(e_m \in Q)$$

$$\leqq \sum_{m=1}^{j-r+1} \frac{K^{r-1}}{(j-m)_{r-1}} \cdot \frac{1}{\binom{j}{m-1}(j-m+1)}$$

$$\leqq \frac{K^{r-1}}{j_r} \cdot \sum_{m=1}^{j-r+1} \frac{(m-1)!(j-m-r+1)!}{(j-r)!} \leqq \frac{K^r}{j_r} \quad \text{by Fact 3.} \quad \square$$

Lemma 4 will show that the probability of a random route intersecting any fixed route r times vanishes exponentially with r increasing. As a preliminary we need to examine an effect of distance in the cube. For $y, z \in V$ the Hamming distance $H(y, z)$ is the number of bits in which the binary representations of y and z differ, i.e.,

$$H(y, z) = |\{i \,|\, y^i \neq z^i\}|.$$

LEMMA 3. *For any $s, t, x \in V$ with $H(s, x) = H(x, t) = k$,*
(A) *in Phase* A

$$\Pr(s \to \textit{through } x) \leqq 1 \Big/ \binom{n}{k},$$

(B) *in Phase B*

$$\Pr(\to t \textit{ through } x) \leqq 1 \Big/ \binom{n}{k}.$$

Proof. (A) The probability that the route from s has length at least k is clearly $B(k, n, \frac{1}{2})$. Now there are $\binom{n}{k}$ nodes at Hamming distance k from s. Assuming that the route from s does have length at least k the probability of it passing through any one of these nodes must be, by symmetry, the same as for any other, namely $1/\binom{n}{k}$. Hence

$$\Pr(s \to \text{through } x) = \Pr(s \to \text{through } x \,||s\to| \geqq k) \cdot \Pr(|s\to| \geqq k) \leqq \frac{1}{\binom{n}{k}} \cdot 1.$$

(B) By a similar argument to the one above:

$$\Pr(\to t \text{ through } x) = \Pr(\to t \text{ through } x \,||t\to| \geqq k) \cdot \Pr(|\to t| \geqq k) \leqq \frac{1}{\binom{n}{k}} \cdot 1. \quad \square$$

LEMMA 4. *For any fixed route R the expected number of packet routes in Phase* A (*and similarly Phase* B) *which have at least r edges in common with R is*

$$A_r = \sum_{s \in V} \Pr(|s \to \cap R| \geqq r) \leqq \min\left\{\frac{n^4 2^r}{n_r}, n\right\}.$$

Proof. We give the proof for Phase A. The analysis for Phase B is identical when the packet movements are played in reverse.

The bound on n is a restatement of the fact observed in Lemma 1 that

$$\sum_{s \in V} p_s < n.$$

Now consider the other bound. Suppose that $R = e_1 e_2 \cdots e_h$ and that it visits $y_0, y_1, \cdots, y_h$ in turn. Let V_{ikg} be the set of nodes such that $H(s, y_i) = k$, and y_{i-g} is the first node of R for which route $s \to y_{i-g} \to y_i$ is possible (i.e., $s \to y_{i-g-1} \to y_i$ is impossible). This is illustrated in Fig. 1.

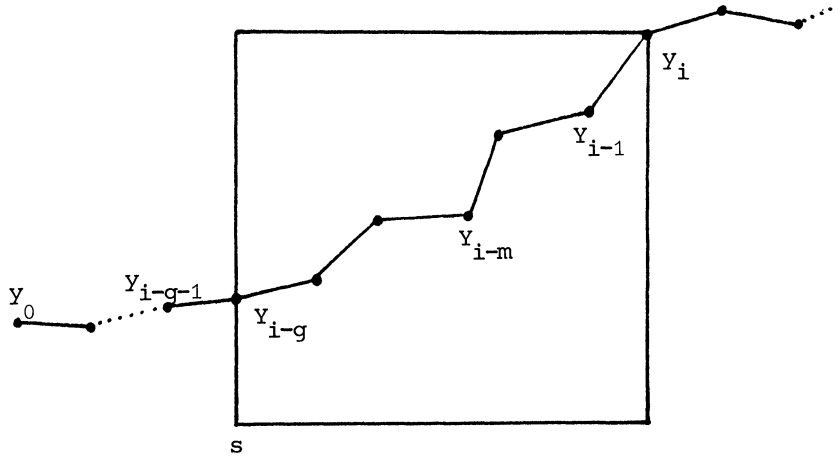

FIG. 1.

Now

$$|V_{ikg}| \leqq \binom{n-g}{k-g} \tag{1}$$

Let $p_{si} = \Pr(|s \to \cap R| \geqq r \,|\, s \to \text{through } y_i)$. Consider now $s \in V_{ikg}$ for some fixed i, k and g, and let y be the first node of R that $s \to$ visits. In this case

$$\begin{aligned} p_{si} &= \sum_{m=r}^{g} \Pr(y = y_{i-m} \,|\, s \to \text{through } y_i) \\ &\quad \cdot \Pr(|s \to \cap e_{i-m+1} \cdots e_i| \geqq r \,|\, y = y_{i-m} \text{ and } s \to \text{through } y_i) \\ &\leqq \sum_{m=r}^{g} \frac{1}{\binom{k}{k-m}} \cdot \frac{K^r}{m_r} \qquad (2) \\ &= \frac{K^r}{k!} \cdot \sum_{m=r}^{g} (k-m)!(m-r)! \leqq \frac{K^r}{k!} \cdot (g-r+1) \cdot (k-r)! \quad \text{since } r \leqq m \leqq g \leqq k. \end{aligned}$$

By definition

$$A_r = \sum_{s \in V} \Pr(|s \to \cap R| \geqq v) \leqq \sum_{i=1}^{h} \sum_{s \in V} p_{si} \cdot \Pr(s \to \text{through } y_i).$$

Since for each i the sets V_{ikg} partition V

$$A_r \leqq \sum_{i=0}^{h} \sum_{k=r}^{n} \sum_{g=r}^{i} \sum_{s \in V_{ikg}} p_{si} \Pr(s \to \text{through } y_i).$$

Using (2) and Lemma 3, we can bound the summand by

$$\frac{K^r}{k!} \cdot (g-r+1) \cdot (k-r)! \cdot \frac{1}{\binom{n}{k}}.$$

By virtue of (1) the sum of the above over V_{ikg} is

$$\frac{K^r}{k!}\cdot\frac{(g-r+1)(k-r)!\cdot k!(n-k)!}{n!}\cdot\frac{(n-g)!}{(k-g)!(n-k)!}$$

$$=\frac{(g-r+1)}{n_r}\cdot\frac{(k-r)_{g-r}}{(n-r)_{g-r}}\cdot K^r\leqq\frac{(g-r)K^r}{n_r}.$$

Hence

$$A_r\leqq\frac{K^r}{n_r}\cdot\sum_{i=0}^{h}\sum_{k=r}^{n}\sum_{g=r}^{i}(g-r+1)\leqq\frac{nh^3K^r}{n_r}$$

as required. □

Although the exponent of n in Lemma 4 can be improved by more careful analysis its value is immaterial unless we wish to study the exact relationship between S and C in the theorem.

Proof of Theorem. We give a proof for Phase A. The argument for B is identical. Consider any route R and a suitably large constant C. The number of edges a route has in common with R we shall call its *overlap r*. In the analysis we deal with overlaps in three different ways depending on which of the following ranges it falls in:

$$[1,4\alpha],\quad\left[4\alpha,\frac{n}{\log_2 n}\right],\quad\left[\frac{n}{\log_2 n},n\right],$$

for an appropriate constant α to be defined. The second range is itself split into about $\log_2 n$ subranges $[2^i,2^{i+1}]$ and C is large enough that

$$\frac{Cn}{24(r-\alpha)(\log_2 n-\alpha)}$$

exceeds unity for all values of r in the second range.

In particular we observe that

$$\begin{aligned}&\Pr\left(\sum_s|s\to\cap R|\geqq Cn\right)\\&\leqq\Pr\left(|s\to\cap R|\geqq 1\text{ for at least }\frac{Cn}{12\alpha}\text{ values of }s\right)\\&\quad+\sum_i\Pr(E_i)+\Pr\left(|s\to\cap R|\geqq\frac{n}{\log_2 n}\text{ for at least }\frac{C}{3}\text{ values of }s\right)\\&=\Pr(E^1)+\Pr(E^2)+\Pr(E^3),\quad\text{say}\end{aligned}\tag{5}$$

where each component E_i of E^2 is itself the event

$$\text{“}|s\to\cap R|\geqq 2^i\text{ for at least }m_i(n)=\frac{Cn}{24(2^i-\alpha)(\log_2 n-\alpha)}\text{ values of }s\text{”}.$$

In order to verify (5) we note that if the overlap sum exceeds Cn then $E^1,\cdots,E_i,E_{i+1},\cdots,E^3$ cannot all be false, for if they were then the contribution from each of the three ranges E^1, E^2 and E^3 would be less than $Cn/3$. In particular, if E^1 is false then the overlap sum in the range $[1,4\alpha]$ is clearly less than $Cn/3$. The same holds for E^3. Finally, if every E_i of E^2 is false also then the overlap sum in the second range

is at most

$$\sum_{i=\lfloor \log_2 4\alpha \rfloor}^{\log_2 n} \frac{Cn}{24(2^i-\alpha)(\log_2 n-\alpha)} \cdot 2^{i+1} \leqq \frac{Cn}{3}$$

for all sufficiently large n.

It remains to show that for a suitable α however large S is chosen there exists C such that the probabilities of E^1, E^2 and E^3 are all bounded by $(\frac{1}{3})2^{-(S+1)n}$. For we can then deduce from (5) that

$$\Pr\left(\sum_s |s \to \cap R| \geqq Cn\right) \leqq 2^{-(S+1)n}.$$

Since there are $N = 2^n$ routes R the probability that some of them do have such a large overlap sum is then at most 2^{-Sn} as required.

That for every α and S $\Pr(E^1) \leqq (\frac{1}{3})2^{-(S+1)n}$ for a suitable C is merely a restatement of Lemma 1 and nothing further needs to be proved.

To bound E^2 it is sufficient to prove the claim that for some C for each i $\Pr(E_i) \leqq (3n)^{-1}2^{-(S+1)n}$. By Fact 2 it suffices to consider the event that at least $m = m_i(n)$ successes occur in N trials with equal probabilities. By Lemma 4 this probability is at most

$$p \leqq \frac{n^4 2^r}{n_r N}, \quad \text{where } r = 2^i,$$

$$\leqq \frac{n^{4-r}2^{2r}}{N}, \quad \text{since } r = 2^i < \frac{n}{2} \text{ in this range.}$$

Using Fact 4

$$B(m, N, p) \leqq (Np)^m e^m \leqq \left(\frac{2^{2r}e}{mn^{r-4}}\right)^m = X, \quad \text{say.}$$

Then

$$\log_2 X \leqq \frac{Cn}{24(r-\alpha)(\log_2 n-\alpha)} \cdot (2r-(r-4)\log_2 n+\log_2 e)$$

$$\leqq \frac{-Cn}{24} \quad \text{if } \alpha = 4 \text{ and } r = 2^i > 4\alpha.$$

We conclude that $\Pr(E_i) \leqq 2^{-Cn/24}$, which is less than $(3n)^{-1}2^{-(S+1)n}$ for C chosen large enough.

Finally to bound E^3 we need an estimate with $r = n/\log_2 n$ and $m = C/3$. Here the analysis for E^2 applies since for $r = n/\log_2 n$

$$\frac{Cn}{24\left(\frac{n}{\log_2 n}-\alpha\right)(\log_2 n-\alpha)} \leqq \frac{C}{3} \quad \text{for all sufficiently large } n.$$

Hence the required bound of $(1/3)2^{-(S+1)n}$ certainly holds for E^3.

We have shown therefore that the overlap sum for the route of each packet is suitably small. Since the overlap sum for a packet bounds the total time it waits in queues the result is established. Although the proof was valid only for sufficiently large values of n, the Theorem can be made to hold for all values of n by always choosing C large enough to cover the remaining small cases. □

5. Remarks.

1. *Testability.* The two phases of the algorithm are testable in the following sense. Phase A is independent of the input altogether. Running it on, say, the identity permutation for different values of F and testing whether it has finished is therefore a method of sampling the distribution of the runtime needed for finishing. In the overall routing algorithm Phase B is used with the packets initially placed randomly in the cube. Hence the distribution of the runtime needed for finishing in Phase B can be sampled by running it with packets placed initially at random. In this way suitable values of F and G can be obtained experimentally. Such experimental results are reported in [11].

2. *Variations.* For practical purposes there is no reason why every packet should wait for Phase A to finish before embarking on Phase B. A modified algorithm in which any packet v immediately enters Phase B as soon as T_v becomes empty in Phase A will also clearly have its runtime bounded by the analysis above, provided the queuing discipline always gives preference to packets still executing Phase A.

Another kind of modification is required if we want the algorithm always to finish. Instead of cutoffs we would have global checks at every n steps to test whether the algorithm has finished. This can be done by collecting one bit of information from each node and collecting their conjunction at one node. For this $O(n)$ steps are sufficient where, now, a step consists of transmitting a fixed piece of bookkeeping along a wire. With this modification each phase is always correct and for all sufficiently large H finishes in time $H \cdot \log_2 N$ with overwhelming probability.

A further variation is possible if we wish to simplify the local computations at the expense of carrying more bookkeeping information. In that case we can precompute the whole of the route for each packet before it leaves its initial node. This also avoids holdups in Phase A that occur whenever $\alpha = 0$ is picked. Note that such holdups occur at most n times for any route and could also be avoided by adding an "until $\alpha = 1$" inner loop to Phase A.

3. *Queuing disciplines.* The proofs given apply to all disciplines, e.g., "first in-first out", "first in-last out". Experimentation appears to be the only method of choosing amongst them. More complicated alternatives include "packets with farthest to go first out".

4. *Obliviousness.* An essential feature of this algorithm is that the route taken by each packet is determined entirely by itself. The other packets can only influence the rate at which the route is traversed. For this reason no global synchronization is required. Indeed, the scheme appears to be well suited to supporting a continuous stream of communication requests from packets generated at the nodes, as long as the traffic flow does not saturate the system either as whole, or by requesting one node or region of it too frequently.

As an alternative "adaptive" algorithms could be considered. For example, one could route the packets from a node so as to minimize the maximal queue length there. Unfortunately such strategies appear to be beyond rigorous analysis or testability.

5. *Necessity for Phase* A. Phase A may appear unnatural at first sight since it may route a packet to distant parts of the network even when its destination is near its source. It is natural to ask therefore whether Phase B on its own works in $O(n)$ steps for *all* inputs, rather than merely for most inputs. A negative answer to this can be derived as follows: Consider any edge $e = (x, y)$. There are $\binom{n-1}{r}$ nodes at distance r from x from which routes can go through e. For each such node z choose as its destination one of the $\binom{n-1}{r}$ nodes at distance r from y to which a route can go from

z via e. The reader can verify that if $r \leqq (n-1)/2$ then this is always possible. Now consider Phase B applied to such a set of $\binom{n-1}{r}$ packets. Since each route will intersect e with probability

$$(r+1)^{-1}\left(\frac{2r+1}{r}\right)^{-1}$$

the expected number of routes intersecting e will be

$$\frac{(n-1)!}{r!(n-r-1)!} \cdot \frac{r!r!}{(2r+1)!} \leqq (r+1)^{-1} \cdot \frac{(n-1)_r}{(2r+1)_r} \leqq (r+1)^{-1}\left(\frac{(n-1)}{(2r+1)}\right)^r$$

If $r = n/J$ this quantity grows as $2^{\gamma n}$ where γ equals $(\log_2 (J/2))/J$, which is positive if $J > 2$. Hence if Phase B is started on a suitably bad input it will require N^γ rather than logarithmic time.

We note that the above estimation is asymptotic. For small values of n it is clearly advantageous to omit Phase A.

6. Alternative algorithms. Recently [12] it has been shown that if in each phase the dimensions are traversed in order of dimension number then the proof of the $(\log N)$ runtime is much simplified, essentially because the 1ntersection of any two routes must then be a contiguous sequence of edges.

REFERENCES

[1] D. ANGLUIN AND L. G. VALIANT, *Fast probabilistic algorithms for hamiltonian circuits and matchings*, J. Comput. System Sci., 18 (1979), pp. 155–193.

[2] K. E. BATCHER, *Sorting networks and their applications*, AFIPS Spring Joint Comp. Conf., 32 (1968), pp. 307–314.

[3] V. E. BENES, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York 1965.

[4] H. CHERNOFF, *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Ann. Math. Statist. 23 (1952), pp. 493–507.

[5] W. HOEFFDING, *On the distribution of the number of successes in independent trials*, Ann. Math. Statist., 27 (1956) pp. 713–721.

[6] G. LEV, N. PIPPENGER AND L. G VALIANT, *A fast parallel algorithm for routing in permutation networks*, IEEE Trans. Comput., C-30 (1981), pp. 93–100.

[7] M. O. RABIN, *Probabilistic algorithms*, Algorithms and Complexity, J. F. Traub, ed., Academic Press, New York, 1976.

[8] H. J. Siegel, *Interconnection networks for* SIMD *machines*, Computer, (June 1979), pp. 57–65.

[9] R. SOLOVAY AND V. STRASSEN, *A fast Monte-Carlo test for primality*, this Journal., 6 (1977), pp. 84–85.

[10] A. WAKSMAN, *A permutation network*, J. Assoc. Comput. Mach., 15 (1968), pp. 159–163.

[11] L. G. VALIANT, *Experiments with a parallel communication scheme*, Proc. 18th Allerton Conf. on Communication, Control and Computing, University of Illinois, Oct. 8–10, 1980, pp. 802–811.

[12] L. G. VALIANT AND G. J. BREBNER, *Universal Schemes for parallel communication*, Proc. 13th ACM Symposium of Theory of Computing, 1981, pp. 263–277.