Transmitting on Various Network Topologies

Tung-Yang Ho and Lih-Hsing Hsu

Department of Information and Computer Science, National Chiao Tung University, Hsinchu, Taiwan 30049, Republic of China

Ting-Yi Sung

Institute of Information Science, Academia Sinica, Taipei, Taiwan 11529, Republic of China

Architectures for interconnection networks can be represented by graphs, while vertices represent processors and edges represent communication links between processors. We use the terms vertices and processors interchangeably. Let $G = (V, E)$ be a graph representing the topology for an interconnection network. Let v_0 be a special vertex outside G, called the *host processor,* which is connected to each vertex in G. The host processor is the sender or source of the message to be transmitted to all of the vertices in G. In each time unit, the host processor may send an identical message to an arbitrary vertex in G or remain idle. At the same time, each processor in G that has already received the message can send it to all its neighbors in one time unit. A transmitting scheme allowing all processors to receive the message in t time units and requiring the host processor to send the message s times is called a (t, s)-transmitting scheme, where $t \geq s$. We call t the transmitting time, and s, the workload of the host processor. We aimed to find an optimal transmitting scheme, i.e., a transmitting scheme such that t is minimized while s is also minimized. In this paper, we present optimal transmitting schemes for linear arrays, rings, complete binary trees, star trees, and (directed) de Bruijn graphs. Furthermore, we present a (t, s)-transmitting scheme for diagonal meshes which are defined slightly different from meshes, and the ratio of t to the optimal transmitting time is approximate to 1.1. *© 1996 John Wiley & Sons, Inc.*

1. INTRODUCTION

As is customary in structure studies of parallel architectures, we restrict our attention to a set of identical processors, and we view the architectures of the underlying interconnection networks as graphs. The vertices of a graph represent the processors of an architecture, and the edges of the graph represent the communication links between processors. Let $G = (V, E)$ be a graph representing the topology for a network. Let v_0 be a special vertex outside G, called the *host processor* (abbreviated as *host*), which is connected to each vertex of V. The host processor is the sender or source of the message to be transmitted to all the vertices in G. In each time unit, the host may send its message to any single vertex of the graph G, according to its choice. At the same time, each processor that has already received the message can send it to all its neighbors in one unit of time. For $u, v \in V$, $d(u, v)$ denotes the shortest distance from u to v. If in the i-th time unit the host sends its message to processor p_i, then after the k-th time unit, $k > i$, all the processors v satisfying $d(p_i, v) \leq k - i$ can receive the message. The objective is to minimize the number of time units such that all the vertices in G can receive the message. This minimum number of time units for graph G is called the *optimal transmitting time* for G, denoted by $t(G)$.

For an n-dimensional hypercube Q_n, Alon [1] showed that $t(Q_n) = \lceil n/2 \rceil + 1$ and gave a simple procedure to achieve this goal. The host processor simply sends its message to an arbitrary processor p_1 in the first time unit, then to the antipodal of p_1, i.e., the vertex having a Hamming distance n from p_1, in the second time unit. Afterward, the host just waits. This transmitting scheme gives the host a rather light workload. We note that even if the

NETWORKS, Vol. 27 (1996) 145–157

CCC 0028-3045/96/020145-13

host sends the message to a processor in each time unit the transmitting time remains the same. Thus, we consider the transmitting problem as a problem to not only determine $t(G)$ but also to minimize the workload of the host when $t(G)$ is achieved. The workload of the host is defined as the number of time units in which the host sends the message to processors in G.

We give here a formal description of the transmitting problem: We are given a graph $G = (V, E)$ and a special vertex v_0 as introduced before. For $v \in V$ and a nonnegative integer r, the *r-neighborhood* of v is defined as $N_r(v) = \{u \in V | d(u, v) \leq r\}$. By convention, $N_0(v) = \{v\}$. Let t be a positive integer and $V' = V \cup \{v_0\}$. We use $[t]$ to denote the set $\{1, 2, \ldots, t\}$. Let f be a function mapping $[t]$ to V' defined as follows: $f(i) = v \neq v_0$ if the host sends its message to v at time i, and $f(i) = v_0$ if the host is idle at time i. Let $s = |\{i | f(i) \in V\}|$, which is called the *workload of the host.* It is obvious that $s \leq t$. The function f is called a *(t, s)-transmitting scheme* if $\cup_{f(i) \in V} N_{t-i}(f(i)) = V$, i.e., all vertices in V can receive the message in t time units. In a (t, s)-transmitting scheme, each processor in G receives the message either from the host or from its neighbor within t time units, while the workload of the host is given by s. Such a t is called a *feasible transmitting time.* The *cost* of a (t, s)-transmitting scheme f, denoted by $c(f)$, is defined as the ordered pair (t, s). Let $TS(G)$ denote the set of all possible transmitting schemes for G. Let $f, g \in TS(G)$ be two transmitting schemes with $c(f) = (t_1, s_1)$ and $c(g) = (t_2, s_2)$. We say $f \preceq g$ if $c(f) \leq c(g)$ lexicographically, i.e., $t_1 < t_2$ or $t_1 = t_2$ and $s_1 \leq s_2$. A scheme $f^* \in TS(G)$ is called an *optimal transmitting scheme* for G if $f^* \preceq g$ for all $g \in TS(G)$. The optimal transmitting cost of G, denoted by $c(G)$, is defined as $c(G) = c(f^*) = (t^*, s^*)$, where $t^* = t(G)$ is called the *optimal transmitting time* and $s^* = s(G)$ is called the *optimal workload of the host.*

Since we can always obtain a $(D + 1, 1)$-transmitting scheme, where D is the diameter of the underlying network, an optimal transmitting scheme is what we seek. The problem of designing an optimal transmitting scheme for a graph G is important to communication of interconnection networks. Alon [1] proposed an optimal $(\lceil n/2 \rceil + 1, 2)$-transmitting scheme for the n-dimensional hypercube Q_n. Besides hypercubes, rings, trees, meshes, and de Bruijn graphs are important topologies for network architectures. In this paper, we give transmitting schemes on rings and some special tree structures, such as linear arrays, complete binary trees, and star trees, and prove their optimality. However, it can be observed that the transmitting problem is difficult for general tree structures. In addition, we propose a function f as defined earlier for diagonal meshes which are defined slightly different from meshes and are a special case of perfect recursive diagonal tori/meshes introduced in [5, 6]. The function f is shown to be a (t, s)-transmitting scheme. The proof already involves a very complicated calculation, not to mention solving the transmitting problem on meshes. Furthermore, we show that the ratio of t to the optimal transmitting time is approximate to 1.1. Finally, the de Bruijn graph is studied since it can interconnect a large number of vertices with small diameter, fixed degree, and recursive construction. The optimal transmitting scheme for a d-ary n-dimensional (directed) de Bruijn graph is presented.

2. TRANSMITTING ON LINEAR ARRAYS AND RINGS

A *linear array* of n vertices, L_n, is an undirected graph given by $V(L_n) = \{v_1, v_2, \ldots, v_n\}$ and $E(L_n) = \{(v_i, v_{i+1}) | 1 \leq i \leq n - 1\}$. A *ring* of length n, C_n, is a graph given by $V(C_n) = V(L_n)$ and $E(C_n) = E(L_n) \cup \{(v_n, v_1)\}$.

Theorem 1. *The optimal transmitting cost of L_n is $(\lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil)$.*

Proof. Without loss of generality, we assume that $n = m^2$ for some positive integer m. Let f be an optimal transmitting scheme with $c(f) = (t, s)$. L_n can be easily partitioned into m nonoverlapping intervals $I_1, I_2, \ldots, I_m$ with $|I_i| = 2i - 1$ vertices for $1 \leq i \leq m$. An (m, m)-transmitting scheme can be easily obtained by setting $f(i)$ to the center of I_{m-i+1}. Thus, $m \geq t$.

On the other hand, it is observed that $|N_r(v)| \leq 2r + 1$ for any vertex v and nonnegative integer r. Since f is a transmitting scheme, it follows that

$$m^2 = |\cup_{f(i) \in V} N_{t-i}(f(i))| \leq \sum_{r=0}^{t-1} (2r + 1)$$
$$= 1 + 3 + \cdots + (2t - 1) = t^2. \quad (1)$$

Thus, $t^2 \geq m^2$, i.e., $t \geq m$. Hence, $t = m$.

Suppose that $s \leq m - 1 = t - 1$. It follows that

$$|\cup_{f(i) \in V} N_{t-i}(f(i))| \leq \sum_{r=1}^{t-1} (2r + 1)$$
$$= 3 + \cdots + (2t - 1) = t^2 - 1 = m^2 - 1,$$

which leads to a contradiction. Consequently, the proposed (m, m)-transmitting scheme is optimal, and, moreover, the theorem follows. ∎

The optimal transmitting cost for C_n can be easily obtained as a corollary.

Corollary 1. *The optimal transmitting cost of C_n is $(\lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil)$.*

3. TRANSMITTING ON COMPLETE BINARY TREES AND FULL BINARY TREES

A *complete binary tree* of height n, denoted by BT_n, is an undirected graph given by $V(BT_n) = \{1, 2, \dots, 2^n - 1\}$, and $(i, j) \in E(BT_n)$ if and only if $\lfloor j/2 \rfloor = i$. The vertex 1 is called the *root* of BT_n. The *level* of the vertex i is defined as $d(1, i) + 1$, which is given by $\lfloor \log_2 i \rfloor + 1$. The *leaves* of BT_n are all of the vertices in level n. We also denote the complete binary tree BT_n as $FT(2^n - 1)$. For a positive integer $j \le 2^n - 1$, the *full binary tree* with j vertices, $FT(j)$, is the subgraph of $FT(2^n - 1)$ induced by vertices $\{1, 2, \dots, j\}$. The *height* of $FT(j)$ is $\lfloor \log_2 j \rfloor + 1$. Two binary trees BT_4 $(= FT(15))$ and $FT(12)$ are illustrated in Figure 1. In the following theorem, we derive the optimal transmitting cost, and the optimal transmitting scheme is given in the proof of the theorem.

Theorem 2. *The optimal transmitting cost for* BT_n *is* $(n, 1)$.

Proof. We give an $(n, 1)$-transmitting scheme by sending the message to the root, vertex 1, of BT_n at the first time unit and then the host simply idles. After n time units, all the vertices will receive the message. Therefore, we only need to show that the message cannot be transmitted to all of the vertices of $BT(n)$ in $n - 1$ time units.

Let f be an $(n - 1, s)$-transmitting scheme. We assume without loss of generality that $s = n - 1$. Let L denote the set of all leaves in BT_n. It follows that $|L| = 2^{n-1}$. For any vertex v in BT_n, it is observed that $|N_{n-1-i}(v) \cap L| \le 2^{n-1-i}$ for $i \le n - 1$ and, moreover, the equality holds if and only if v is at level $i + 1$. Since f is an $(n - 1, n - 1)$-transmitting scheme, $L = \bigcup_{f(i) \in V}(N_{n-1-i}(f(i)) \cap L)$. However, $|\bigcup_{f(i) \in V}(N_{n-1-i}(f(i)) \cap L)| \le \sum_{i=1}^{n-1} 2^{n-1-i} = 2^{n-1} - 1 < |L|$, which is a contradiction. Hence, the proposed $(n, 1)$-transmitting scheme is optimal. The theorem follows. ■

Observe from Figure 1 that $FT(12)$ is a subgraph of BT_4. There exists a trivial $(4, 1)$-transmitting scheme for $FT(12)$. However, we have a $(3, 3)$-transmitting scheme for $FT(12)$. For example, the host sends its message to vertex 2 in the first time unit, then to vertex 6 at the second time unit, and, finally, to vertex 7 at the third time unit. It can be easily verified that all the vertices in $FT(12)$ receive the message in three time units. Using similar arguments in the proof of Theorem 2, we can obtain the following corollary:

Corollary 2. *For* $1 \le m \le 2^{n-1}$, *the optimal transmitting cost of* $FT(2^n - m)$ *is* $(n, 1)$ *if* $m = 1, 2$, *and* $(n - 1, n - \lfloor \log_2(m - 1) \rfloor)$ *otherwise.*

Optimal transmitting schemes for $FT(2^n - m)$ are given as follows: When $m = 1, 2$, an optimal $(n, 1)$-transmitting scheme f is given by $f(1) = 1$ and $f(i) = v_0$ for all $2 \le i \le n$. When $2^{n-k} + 1 \le m \le 2^{n-k+1}$ and $3 \le k \le n - 1$, an optimal $(n - 1, k)$-transmitting scheme is given by $f(1) = 2$, $f(i) = 2(f(i - 1) + 1)$ for all $2 \le i \le k - 1$, and $f(k) = f(k - 1) + 1$. When $2^{n-2} + 1 \le m \le 2^{n-1}$, an optimal $(n - 1, 2)$-transmitting scheme is given by $f(1) = 2, f(2) = 3$.

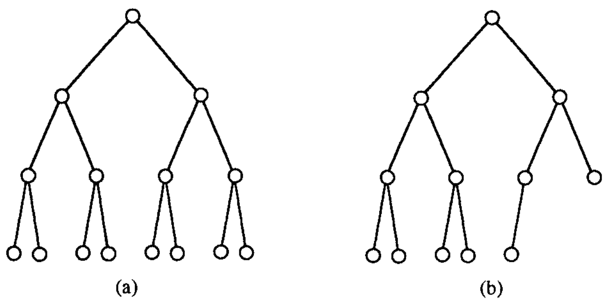

Fig. 1. (a) Complete binary tree $BT_4(=FT(15))$; (b) Full binary tree with 12 vertices $FT(12)$.

4. TRANSMITTING ON STAR TREES

Let P_n^k denote the graph which is the disjoint union of k linear arrays, each of n vertices, i.e., $V(P_n^k) = \{v_i^j \mid 1 \le i \le n, 1 \le j \le k\}$ and $E(P_n^k) = \{v_i^j, v_{i+1}^j) \mid 1 \le i \le n - 1, 1 \le j \le k\}$. For an integer $k \ge 2$, the *star tree* S_n^k is the graph given by $V(S_n^k) = V(P_n^k) \cup \{u_o\}$ with $u_0 \notin V(P_n^k)$ and $E(S_n^k) = E(P_n^k) \cup \{(u_o, v_1^j) \mid 1 \le j \le k\}$. The following number theory result is required to evaluate the transmitting time of S_n^k for some n and k:

Lemma 1. *Let* p *and* n *be positive integers such that* $p \mid n$ *and* $p \ne n$. *The set* O_n, *consisting of all positive odd integers less than or equal to* $2n - 1$, *can be partitioned into* p *disjoint subsets,* $D_1, D_2, \dots, D_p$, *such that the sum of the elements in* D_i *equals* n^2/p *for* $1 \le i \le p$.

Proof. We prove this lemma by the construction of $D_1, D_2, \dots, D_p$. Let $q = n/p$. Consider the following cases:

CASE 1. q is an even integer.
Let $D_k^j = \{4(j - 1)p + 2k - 1, 4jp - (2k - 1)\}$ for $1 \le k \le p$ and $1 \le j \le q/2$. For a fixed j, $\{D_k^j \mid 1 \le k \le p\}$ forms a partition of the set $\{l \mid 4(j - 1)p + 1 \le l \le 4jp - 1\}$, and the sum of the elements in D_k^j is $(8j - 4)p$. Define $D_k = \bigcup_{j=1}^{q/2} D_k^j$. Then, the set $\{D_k \mid 1 \le k \le p\}$ forms a partition of the set O_n such that $\sum_{x \in D_k} x = \sum_{j=1}^{q/2} (8j - 4)p = pq^2 = n^2/p$.

CASE 2. $q = 3$.
It follows that $n^2/p = (3p)^2/p = 9p$. Now, we consider two possibilities of p:

SUBCASE 2.1. p is an even integer.
Let $D_i = \{3p + (2i - 1), 6p - (2i - 1)\}$ for $1 \le i \le p/2$. It is easy to see that $\{D_i | 1 \le i \le p/2\}$ forms a partition of the set $\{2i - 1 | 3p + 1 \le 2i - 1 \le 4p - 1 \text{ or } 5p + 1 \le 2i - 1 \le 6p - 1\}$. Moreover, the sum of all elements in D_i is $9p$ for $1 \le i \le p/2$. Let $D_{i+p/2} = \{2i - 1, 2p - 2i + 1, 2p + 2i - 1, 5p - 2i + 1\}$. It can be easily verified that $\{D_{i+p/2} | 1 \le i \le p/2\}$ forms a partition of the set $\{2i - 1 | 1 \le 2i - 1 \le 3p - 1 \text{ or } 4p + 1 \le 2i - 1 \le 5p - 1\}$. Moreover, the sum of all elements in $D_{i+p/2}$ is $9p$ for every $1 \le i \le p/2$. Hence, the set $\{D_k | 1 \le k \le p\}$ forms a partition of the set O_n such that $\sum_{x \in D_k} x = n^2/p$.

SUBCASE 2.2. p is an odd integer.
For $1 \le i \le p$, let D_i be $\{2i - 1, 2p + 2\lceil p/2 \rceil - i, 6p - i\}$ if i is an odd integer, and $\{2i - 1, 4p - i + 1, 4p + 2\lceil p/2 \rceil - i - 1\}$ otherwise. It is easy to verify that $\{D_k | 1 \le k \le p\}$ forms a partition of O_n such that $\sum_{x \in D_k} x = n^2/p$.

CASE 3. q is an odd integer and $q \ge 5$.
Applying the result in Case 2, we can decompose the set O_{3p} into p subsets, $D'_1, D'_2, \ldots, D'_p$, such that the sum of all elements in each subset equals $9p^2/p = 9p$. Let

$$D_1 = D'_1 \cup \{6p + 1, 10p - 1, 10p + 1, 14p - 1, \ldots, 2(q - 2)p + 1, 2pq - 1 = 2n - 1\},$$
$$D_2 = D'_2 \cup \{6p + 3, 10p - 3, 10p + 3, 14p - 3, \ldots, 2(q - 2)p + 3, 2pq - 3 = 2n - 3\},$$
$$D_3 = D'_3 \cup \{6p + 5, 10p - 5, 10p + 5, 14p - 5, \ldots, 2(q - 2)p + 5, 2pq - 5 = 2n - 5\},$$
$$\vdots$$
$$D_p = D'_p \cup \{8p - 1, 8p + 1, 12p - 1, 12p + 1, \ldots, 2(q - 1)p - 1, 2(q - 1)p + 1\}.$$

It can be easily verified that $\{D_k | 1 \le k \le p\}$ forms a partition of O_n. Moreover, the sum of all elements in each D_k is $9p + 16p + 24p + \cdots + 4(q - 1)p = n^2/p$.

Hence, the lemma follows. ∎

Based on the above lemma, we have the following results:

Theorem 3. *Let p and n be positive integers such that $p | n$ and $p \neq n$. Then, the optimal transmitting costs of $P^p_{n^2/p}$ and $S^p_{n^2/p+n}$ are (n, n) and $(n + 1, n + 1)$, respectively.*

Theorem 4. *The optimal transmitting cost of S^p_k is less than or equal to $(\lceil px \rceil + 1, \lceil px \rceil + 1)$, where $x = (-p + \sqrt{p^2 + 4pk})/(2p)$, and equality holds when px is an integer.*

Proof. It is easy to see that S^p_k is a subgraph of $S^p_{(p^2x^2/p)+px}$, where x is the smallest positive integer satisfying $(p^2x^2/p) + px \ge k$. This theorem follows from Theorem 3. ∎

5. TRANSMITTING ON DIAGONAL MESHES

A *diagonal mesh* of *size n*, denoted by M_n, is an undirected graph with vertex set $V = \{(i, j) | 1 \le i, j \le n\}$ and edge set $E = \{((i, j), (i', j')) | |i - i'| \le 1 \text{ and } |j - j'| \le 1\}$. An example of M_5 is shown in Figure 2. This definition is slightly different from the conventional definition for meshes. The diagonal mesh is a special case of perfect recursive diagonal tori/meshes [5, 6]. Given a (conventional) mesh M, a *perfect recursive diagonal mesh* (*PRDM*), denoted by PRDM(d, r), where d and r are positive integers, is defined as $\cup_{i=0}^{r} M^i$, where $M^0 = M$, and M^i is constructed in the following way: Each vertex (x, y) in M^i is connected to vertices (x^i, y^i) where

$$(x^i, y^i) = \begin{cases} (x \pm id, y \pm id) & \text{if } i \text{ is odd,} \\ (x \pm id, y) \text{ and } (x, y \pm id) & \text{if } i \text{ is even.} \end{cases}$$

By convention, if $x \pm id \notin \{1, 2, \ldots, n\}$ or $y \pm id \notin \{1, 2, \ldots, n\}$, the corresponding vertex becomes vacuous. PRDM(d, r) has a maximum degree $4(r + 1)$. In [5, 6], the authors proposed and studied an architecture called *recursive diagonal mesh* (*RDM*) which is a subgraph of PRDM(d, r) containing M^0. Each RDM is constructed from PRDM under a specified selection policy. *Perfect recursive diagonal torus* (*PRDT*) and *recursive diagonal torus* (*RDT*) are similarly constructed as PRDM and RDM, respectively. The RDM/RDT architectures can achieve a small diameter with a reasonable degree and can emulate hypercubes and trees easily; refer to [5, 6] for details. Based on the RDT, a massively parallel machine has been under development in the Japan University

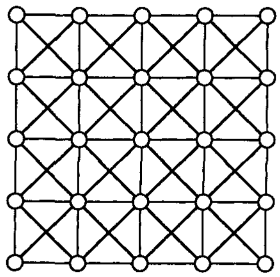

Fig. 2. M_5, a diagonal mesh of size 5.

Massively Parallel Processing Project. The diagonal mesh defined in this paper is indeed a PRDM(1, 1).

We use meshes to mean diagonal meshes for convenience. The vertices (i, j) for $i, j = 1$ or n are called the *boundary* of M_n. To be precise, the vertices (i, j) are called the top, bottom, left, and right boundaries, which are also called *one-side boundaries*, if $i = 1$, $i = n$, $j = 1$, and $j = n$, respectively. A *two-side boundary* is the union of two connected one-side boundaries. There are four two-side boundaries, called the left-top, right-top, left-bottom, and right-bottom boundaries. A mesh is said to be laid on a one-side (say, top) boundary of M_n if the mesh is laid inside M_n with its one-side (top) boundary coinciding with the one-side (top) boundary of M_n. Similarly, we define "a mesh laid on a two-side boundary."

In this section, we give an approximation algorithm to find a transmitting scheme on M_n. When a processor v receives the message, all the processors in $N_r(v)$ can receive the message after r time units. Without loss of generality, we assume that $N_r(v)$ is a mesh M_{2r+1} which contains $(2r + 1)^2$ vertices. A processor p_i receiving the message from the host at the i-th time unit can transmit the message to the processors in $N_{k-i}(p_i)$, a mesh $M_{2(k-i)+1}$, at the k-th time unit where $k \geq i \geq 1$. Therefore, finding the optimal transmitting time is equivalent to finding the smallest integer t such that t meshes $M_1, M_3, \ldots, M_{2t-1}$ can cover M_n. The transmitting strategy is to send the message from the host to the center vertices of these meshes.

Let t be a feasible transmitting time. It follows that t satisfies the following constraint on the number of vertices:

$$\sum_{r=1}^{t} (2r-1)^2 = \frac{4}{3}t^3 - \frac{t}{3} \geq n^2, \qquad (2)$$

i.e., $t^3 - \frac{t}{4} \geq \frac{3}{4} n^2$. To approximate t and ensure the feasibility of t, we choose t to satisfy the constraint

$$t^3 \geq n^2. \qquad (3)$$

In other words, if t satisfies (3), it follows that t also satisfies (2). Note that if t is a feasible transmitting time, t' is obviously also a feasible transmitting time when $t' > t$. We give a transmitting scheme with transmitting time $t = \lceil n^{2/3} \rceil + 2$. The "+2" term in the choice of t is added to satisfy some special cases of n, whereas $t = \lceil n^{2/3} \rceil$ is feasible for most cases of n. We restrict the following discussion to $t = \lceil n^{2/3} \rceil$ is feasible for most cases of n. We restrict the following discussion to $t = \lceil n^{2/3} \rceil$ for most cases of n and consider the special cases of n later.

Our transmitting strategy is as follows:

Choice of t: Choose $t = \lceil n^{2/3} \rceil$.

Mesh arrangement: Let m be the smallest number of meshes required to be laid on each one-side boundary of M_n such that all vertices of the boundary of M_n are covered by $4m - 4$ meshes, $M_{2t-1}, M_{2t-3}, \ldots, M_{2t-8m+9}$. These meshes are laid on the one-side or two-side boundaries of M_n in the following sequence: (left-top, right-bottom, right-top, left-bottom), (bottom, right, left, top), (top, left, right, bottom), (bottom, right, left, top), and so on. (Parentheses in the sequence are added for clarity of the pattern.) This arrangement is illustrated in Figure 3. Once we have m meshes arranged on each one-side boundary, we can reduce the problem from M_n to a mesh of smaller size.

Reduction step: Consider the last four meshes arranged by the above strategy. If the mesh $M_{2t-8m+15}$ is arranged on the top boundary of M_n, it follows that $M_{2t-8m+13}$, $M_{2t-8m+11}$, and $M_{2t-8m+9}$ are arranged on the left, right, and bottom boundaries of M_n, respectively. On the other hand, if the mesh $M_{2t-8m+15}$ is arranged on the bottom boundary of M_n, then $M_{2t-8m+13}$, $M_{2t-8m+11}$, and $M_{2t-8m+9}$ are arranged on the right, left, and top boundaries of M_n, respectively. In either case, the uncovered portion of M_n can be covered by $M_{\hat{n}}$, where

$$\begin{aligned} \hat{n} &= n - (2t - 8m + 11) - (2t - 8m + 13) \\ &= n - (2t - 8m + 9) - (2t - 8m + 15) \\ &= n - 4t + 16m - 24. \end{aligned}$$

Thus, we reduce the problem from M_n to $M_{\hat{n}}$ and the transmitting time from t to $t - 4m + 4$.

To show the correctness of the transmitting strategy, it requires the following lemma to show the derivation of m so that $\hat{n}$ can be determined in the reduction step:

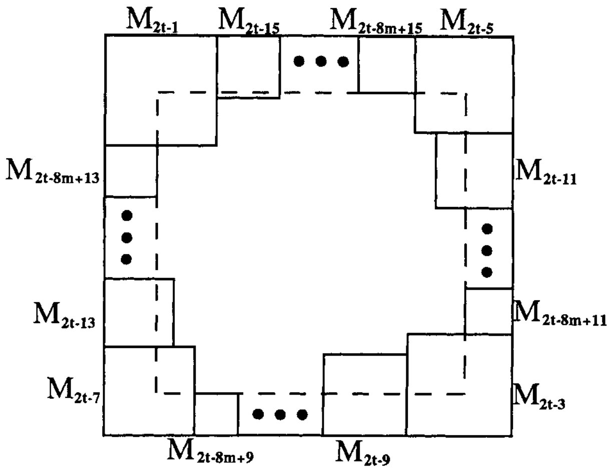

Fig. 3. Mesh arrangement.

Lemma 2. *Let m be the number of meshes laid on each one-side boundary of M_n. By the above arrangement strategy, when m is even, the sums of sizes of meshes laid on the top, left, right, and bottom boundaries in M_n are $2tm - (4m^2 - 8m + 6)$, $2tm - (4m^2 - 8m + 8)$, $2tm - (4m^2 - 8m + 8)$, and $2tm - (4m^2 - 8m + 10)$, respectively. When m is odd and greater than one, these sums are $2tm - (4m^2 - 8m + 9)$, $2tm - (4m^2 - 8m + 9)$, $2tm - (4m^2 - 8m + 7)$, and $2tm - (4m^2 - 8m + 7)$, respectively.*

Proof. We prove the lemma by induction. In this proof, we state "the sums" to mean "the sums of sizes of meshes laid on the top, left, right, and bottom boundaries" for convenience. When $m = 2$, it is the case that only four meshes M_{2t-1}, M_{2t-3}, M_{2t-5}, and M_{2t-7} are laid on each two-side boundary of M_n. The sums are $4t - 6$, $4t - 8$, and $4t - 10$, respectively, which equal $2t \times 2 - (4 \times 2^2 - 8 \times 2 + 6)$, $2t \times 2 - (4 \times 2^2 - 8 \times 2 + 8)$, $2t \times 2 - (4 \times 2^2 - 8 \times 2 + 8)$, and $2t \times 2 - (4 \times 2^2 - 8 \times 2 + 10)$.

When $m = 3$, the sums are $6t - 21$, $6t - 21$, $6t - 19$, and $6t - 19$, respectively, which equal $2t \times 3 - (4 \times 3^2 - 8 \times 3 + 9)$, $2t \times 3 - (4 \times 3^2 - 8 \times 3 + 9)$, $2t \times 3 - (4 \times 3^2 - 8 \times 3 + 7)$, and $2t \times 3 - (4 \times 3^2 - 8 \times 3 + 7)$.

Assume that $m = k$ and that the sums are, respectively, $2tk - (4k^2 - 8k + 6)$, $2tk - (4k^2 - 8k + 8)$, $2tk - (4k^2 - 8k + 8)$, and $2tk - (4k^2 - 8k + 10)$ if k is even and are, respectively, $2tk - (4k^2 - 8k + 9)$, $2tk - (4k^2 - 8k + 9)$, $2tk - (4k^2 - 8k + 7)$, and $2tk - (4k^2 - 8k + 7)$ if k is odd and greater than one. When $m = k$, $4k - 4$ meshes have been arranged accordingly.

Now we consider $m = k + 1$. We first consider that m is even. It follows that k is odd and greater than one. The first $4k - 4$ meshes have been arranged according to the mesh arrangement strategy. Furthermore, the last four meshes, $M_{2t-8k+7}$, $M_{2t-8k+5}$, $M_{2t-8k+3}$, and $M_{2t-8k+1}$, are laid on the top, left, right, and bottom boundaries, respectively. It follows from the induction assumption that the sum of sizes of $k + 1$ meshes laid on the top boundary is given by

$$\begin{aligned} &2tk - (4k^2 - 8k + 9) + (2t - 8k + 7) \\ &= t(k + 1) - (4k^2 + 2) \\ &= 2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 6). \end{aligned}$$

Similarly, we can show the sums of sizes of $k + 1$ meshes laid on the left, right, and bottom boundaries are given by $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 8)$, $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 8)$, and $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 10)$, respectively. We can similarly prove that when $m = k + 1$ is odd the sums are $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 9)$, $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 9)$, $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 7)$, and $2t(k + 1) - (4(k + 1)^2 - 8(k + 1) + 7)$, respectively. Thus, the lemma follows. ∎

It follows from Lemma 2 that the smallest sum of sizes of m meshes on the four one-side boundaries in M_n is $2tm - (4m^2 - 8m + 10)$ when m is even and $2tm - (4m^2 - 8m + 9)$ when m is odd and greater than one. This also implies that at least $2tm - (4m^2 - 8m + 10)$ vertices of each one-side boundary are covered by these $4m - 4$ meshes. Thus, by the definition of m, we choose m to be an integer as small as possible and satisfying

$$\begin{aligned} &2tm - (4m^2 - 8m + 10) \geq n, \\ &\text{i.e., } 4m^2 - (8 + 2t)m + (10 + n) \leq 0. \end{aligned} \tag{4}$$

Once m is determined, the problem is reduced from M_n to a smaller mesh $M_{n-4t+16m-24}$, and the transmitting time is reduced from t to $t - 4m + 4$ as shown in the reduction step. Hence, the reduction step is correct.

Let $t' = n^{2/3}$ and $t = \lceil t' \rceil \geq t'$. It implies that $m'' \geq m'$, where m'' and m' are the smaller roots of $2t'm - (4m^2 - 8m + 10) - n = 0$ and $2tm - (4m^2 - 8m + 10) - n = 0$, respectively. Nonetheless, it suffices to choose $m = \lceil m' \rceil$, rather than $\lceil m'' \rceil$, as shown in (6) of the proof of Lemma 3. Based on the choice of t and m, we show the following lemma to ensure the feasibility of the reduction step in the proposed transmitting strategy as specified in terms of (3):

Lemma 3. *When n is large enough, we have*

$$(t - 4m + 4)^3 \geq (n - 4t + 16m - 24)^2. \tag{5}$$

Proof. Let t, t', m', and m'' be defined as above. Since $t \geq t'$, $m' + 1 \geq m$, and $m'' \geq m'$, it follows that

$$(t - 4m + 4)^3 \geq (t' - 4m')^3 \geq (t' - 4m'')^3, \tag{6}$$

and $(n - 4t' + 16m'' - 8)^2$

$$\geq (n - 4t' + 16m' - 8)^2 \geq (n - 4t + 16m - 24)^2.$$

Thus, it suffices to show that

$$(t' - 4m'')^3 \geq (n - 4t' + 16m'' - 8)^2. \tag{7}$$

Let $k = n^{1/3}$. We thus have $t' = k^2$ and $m'' = [(4 + k^2) - \sqrt{(4 + k^2)^2 - 4(k^3 + 10)}]/4$. Furthermore, we have

$$\begin{aligned} (t' - 4m'')^3 &= [k^2 - (4 + k^2) + \sqrt{(4 + k^2)^2 - 4(k^3 + 10)}]^3 \\ &= (k^4 + 8k^2 - 4k^3 + 24) \\ &\quad \times \sqrt{(4 + k^2)^2 - 4(k^3 + 10)} \\ &\quad - 12k^4 + 48k^3 - 96k^2 + 224, \end{aligned}$$

and

$$(n - 4t' + 16m'' - 8)^2$$
$$= (k^3 - 4k^2 + 4(4 + k^2) - 4\sqrt{(4 + k^2)^2 - 4(k^3 + 10)} - 8)^2$$
$$= k^6 + 16k^4 - 48k^3 + 128k^2 - 320 - (8k^3 + 64)\sqrt{(4 + k^2)^2 - 4(k^3 + 10)}.$$

Let $f(k) = (t' - 4m'')^3 - (n - 4t' + 16m'' - 8)^2$. It follows that

$$f(k) = (k^4 + 4k^3 + 8k^2 + 88) \times \sqrt{(4 + k^2)^2 - 4(k^3 + 10)} - (k^6 + 28k^4 - 96k^3 + 224k^2 - 544).$$

Since $\lim_{k\to\infty} \sqrt{(4 + k^2)^2 - 4(k^3 + 10)}/k^2 = 1$, it follows that $\lim_{k\to\infty} [f(k)]/k^5 = O(1)$. Thus, $f(k) = O(k^5)$. It means that there exists a large number M such that $f(k) \geq 0$ for all $k \geq M$. Hence, the inequality (7) is satisfied for large n. Consequently, the lemma follows. ■

Based on a detailed calculation in the proof of Lemma 3, it can be shown that for $n \geq 729$ we have $(t' - 4m'')^3 \geq (n - 4t' + 16m'' - 8)^2$, and, obviously, (5) is satisfied. However, satisfaction of (5) for all n is to be sought. With the aid of a computer program, it can be verified that (5) is satisfied for $n \geq 397$ when $t = \lceil n^{2/3} \rceil$ and m is the smallest integer that satisfies $4m^2 - (8 + 2t)m + (10 + n) \leq 0$. Thus, we focus on $n \leq 396$ since reduction can be applied for $n \geq 397$. We also note that (5) is satisfied for most cases of $1 \leq n \leq 396$. In other words, choosing $t = \lceil n^{2/3} \rceil$ for M_n, then $t - 4m + 4$ is a feasible transmitting time for $M_{n-4t+16m-24}$ as well, except for some special cases. Examining the computer program output shown in Table I, the set of X of special cases of n that (5) cannot be satisfied is given as follows: $X = \{8, 11, 35{:}52, 55{:}58, 63, 64, 148{:}164, 167{:}172, 178{:}181, 189, 383{:}385, 395, 396\}$, where "$a{:}b$" represents $a, a + 1, \ldots, b$. We define

$$Q = X - \{8, 11, 35{:}45\}. \qquad (8)$$

To eliminate the violation of (5) for these cases of n, we choose $t = \lceil n^{2/3} \rceil + 2$ instead of $t = \lceil n^{2/3} \rceil$. The intuition is to increase the number of meshes to cover M_n and to enlarge the largest mesh used to cover M_n. We illustrate the intuition why we need "+2" in the definition of t by the following example:

Consider $n = 163 \in Q$, and let $t = \lceil n^{2/3} \rceil$. Then, we have $t = 30$ and $m = 4$. Once m is determined, the remaining mesh is of size 83 and the remaining time is 18 for M_{83}. But we have $(18)^3 < (83)^2$, a violation of (5). We thus increase t by 2 to 32. Though the increase of t may decrease m, we choose m to be the same as before, i.e., when $t = 32$, we still have $m = 4$. We put the meshes $M_{63}, M_{61}, \ldots, M_{41}$ in the same places as in the case of $t = 30$. Choosing $t = 32$, the remaining mesh has a size less than or equal to 83 and the remaining transmitting time is 20. Since $(20)^3 \geq (83)^2$, $t = 20$ is a feasible transmitting time for M_{83}.

Based on the above discussion, we present our transmitting strategy in the following algorithm to find a transmitting scheme for M_n with transmitting time $\lceil n^{2/3} \rceil$ or $\lceil n^{2/3} \rceil + 2$, when $n \geq 46$. For $n \leq 45$, we manually solve this transmitting problem with $t = \lceil n^{2/3} \rceil + 1$ for $n = 8$, 11, and $t = \lceil n^{2/3} \rceil$ otherwise; for an example, the solution for M_{45} is illustrated in Figure 4.

Algorithm $TA(n, t)$//Initial value of t is $-\infty$//

(1) If $n \leq 0$, STOP.
(2) If $t = -\infty$, calculate $t = \lceil n^{2/3} \rceil$. Moreover, if $n \leq 45$, we manually solve this transmitting problem with $t = \lceil n^{2/3} \rceil + 1$ for $n = 8$, 11, and $t = \lceil n^{2/3} \rceil$ otherwise. Then STOP.
(3) If $n \leq 45$, we manually solve this transmitting problem, and STOP.
(4) If $2t - 1 \geq n$, send a message to the processor at the center of M_n, and STOP.
(5) Calculate $m = \lceil [(4 + t) - \sqrt{(4 + t)^2 - 4(n + 10)}]/4 \rceil$ and $\hat{n} = n - 4t + 16m - 24$.
(6) If $n \in Q$, set t to be $t + 2$.
(7) Arrange $M_{2t-1}, M_{2t-3}, \ldots, M_{2t-8m+9}$ into M_n according to the mesh arrangement strategy specified before.
(8) $\hat{t} = t - 4m + 4$; call $TA(\hat{n}, \hat{t})$.

A question may naturally arise whether or not the size of the remaining mesh $\hat{n}$ will fall in Q again when we choose the transmitting time as $t = \lceil n^{2/3} \rceil + 2$ for $n \in Q$. If yes, it implies the failure of induction on a reduction step based on Lemma 3, and, furthermore, the transmitting time can be $\lceil n^{2/3} \rceil + \alpha$, where α is a large constant. Since $\alpha = 0$ or 2 is what we seek, we examine Table I more closely and have the following observations to ensure this:

Observations:

(1) For $n \in Q$, the size of the remaining mesh $\hat{n}$ will never fall in Q again.
(2) For $n \notin Q$, the size of the remaining mesh $\hat{n}$ may fall in Q. But we have $\hat{t} \geq \lceil \hat{n}^{2/3} \rceil + 2$, which ensures the feasibility of the reduced problem on the mesh of size $\hat{n}$. Thus, we do not need to increase $\hat{t}$, and t is still feasible.

TABLE I. Computer output for calculation of *t*, *m*, and $\hat{t}$ when n ≤ 732.

n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$
									n = 1–244										
1	1	—	—	—	62	16	3	22	8	123	25	3	47	17	184	33	4	92	21
2	2	—	—	—	63	16 + 1	3	23	8 + 1	124	25	3	48	17	185	33	4	93	21
3	3	—	—	—	64	16 + 1	3	24	8 + 1	125	25	3	49	17	186	33	4	94	21
4	3	—	—	—	65	17	3	21	9	126	26	3	46	18	187	33	4	95	21
5	3	—	—	—	66	17	3	22	9	127	26	3	47	18	188	33	4	96	21
6	4	—	—	—	67	17	3	23	9	128	26	3	48	18	189	33 + 1	4	97	21 + 1
7	4	—	—	—	68	17	3	24	9	129	26	3	49	18	190	34	4	94	22
8	4 + 1	2	—	—	69	17	3	25	9	130	26	3	50	18	191	34	4	95	22
9	5	—	—	—	70	17	3	26	9	131	26	3	51	18	192	34	4	96	22
10	5	2	—	—	71	18	3	23	10	132	26	3	52	18	193	34	4	97	22
11	5 + 1	2	—	—	72	18	3	24	10	133	27	3	49	19	194	34	4	98	22
12	6	2	—	—	73	18	3	25	10	134	27	3	50	19	195	34	4	99	22
13	6	2	—	—	74	18	3	26	10	135	27	3	51	19	196	34	4	100	22
14	6	2	—	—	75	18	3	27	10	136	27	3	52	19	197	34	4	101	22
15	7	2	—	—	76	18	3	28	10	137	27	3	53	19	198	34	4	102	22
16	7	2	—	—	77	19	3	25	11	138	27	3	54	19	199	35	4	99	23
17	7	2	—	—	78	19	3	26	11	139	27	3	55	19	200	35	4	100	23
18	7	2	—	—	79	19	3	27	11	140	27	3	56	19	201	35	4	101	23
19	8	2	—	—	80	19	3	28	11	141	28	3	53	20	202	35	4	102	23
20	8	2	—	—	81	19	3	29	11	142	28	3	54	20	203	35	4	103	23
21	8	2	—	—	82	19	3	30	11	143	28	3	55	20	204	35	4	104	23
22	8	2	—	—	83	20	3	27	12	144	28	3	56	20	205	35	4	105	23
23	9	2	—	—	84	20	3	28	12	145	28	3	57	20	206	35	4	106	23
24	9	2	—	—	85	20	3	29	12	146	28	3	58	20	207	35	4	107	23
25	9	2	—	—	86	20	3	30	12	147	28	3	59	20	208	36	4	104	24
26	9	2	—	—	87	20	3	31	12	148	28 + 2	4	76	16 + 2	209	36	4	105	24
27	9	2	—	—	88	20	3	32	12	149	29 + 1	4	73	17 + 1	210	36	4	106	24
28	10	2	—	—	89	20	3	33	12	150	29 + 1	4	74	17 + 1	211	36	4	107	24
29	10	2	—	—	90	21	3	30	13	151	29 + 1	4	75	17 + 1	212	36	4	108	24
30	10	2	—	—	91	21	3	31	13	152	29 + 1	4	76	17 + 1	213	36	4	109	24
31	10	2	—	—	92	21	3	32	13	153	29 + 2	4	77	17 + 2	214	36	4	110	24
32	11	2	—	—	93	21	3	33	13	154	29 + 2	4	78	17 + 2	215	36	4	111	24
33	11	2	—	—	94	21	3	34	13	155	29 + 2	4	79	17 + 2	216	36	4	112	24
34	11	2	—	—	95	21	3	35	13	156	29 + 1	4	80	17 + 2	217	37	4	109	25
35	11	3	15	3	96	21	3	36	13	157	30 + 1	4	77	18 + 1	218	37	4	110	25
36	11	3	12	4	97	22	3	33	14	158	30 + 1	4	78	18 + 1	219	37	4	111	25
37	12	3	13	4	98	22	3	34	14	159	30 + 1	4	79	18 + 1	220	37	4	112	25
38	12	3	14	4	99	22	3	35	14	160	30 + 1	4	80	18 + 1	221	37	4	113	25
39	12	3	15	4	100	22	3	36	14	161	30 + 1	4	81	18 + 1	222	37	4	114	25
40	12	3	16	4	101	22	3	37	14	162	30 + 1	4	82	18 + 1	223	37	4	115	25
41	12	3	13	5	102	22	3	38	14	163	30 + 2	4	83	18 + 2	224	37	4	116	25
42	13	3	14	5	103	22	3	39	14	164	30 + 2	4	84	18 + 2	225	37	4	117	25
43	13	3	15	5	104	23	3	36	15	165	31	4	81	19	226	38	4	114	26
44	13	3	16	5	105	23	3	37	15	166	31	4	82	19	227	38	4	115	26
45	13	3	17	5	106	23	3	38	15	167	31 + 1	4	83	19 + 1	228	38	4	116	26
46	13 + 2	3	18	5 + 2	107	23	3	39	15	168	31 + 1	4	84	19 + 1	229	38	4	117	26
47	14 + 1	3	15	6 + 1	108	23	3	40	15	169	31 + 1	4	85	19 + 1	230	38	4	118	26
48	14 + 1	3	16	6 + 1	109	23	3	41	15	170	31 + 1	4	86	19 + 1	231	38	4	119	26
49	14 + 1	3	17	6 + 1	110	23	3	42	15	171	31 + 1	4	87	19 + 1	232	38	4	120	26
50	14 + 2	3	18	6 + 2	111	24	3	39	16	172	31 + 1	4	88	19 + 1	233	38	4	121	26
51	14 + 2	3	19	6 + 2	112	24	3	40	16	173	32	4	85	20	234	38	4	122	26
52	14 + 2	3	20	6 + 2	113	24	3	41	16	174	32	4	86	20	235	39	4	119	27
53	15	3	17	7	114	24	3	42	16	175	32	4	87	20	236	39	4	120	27

Table I continues

TABLE I. Continued

n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$
									n = 1–244 (Continued)										
54	15	3	18	7	115	24	3	43	16	176	32	4	88	20	237	39	4	121	27
55	15 + 1	3	19	7 + 1	116	24	3	44	16	177	32	4	89	20	238	39	4	122	27
56	15 + 1	3	20	7 + 1	117	24	3	45	16	178	32 + 1	4	90	20 + 1	239	39	4	123	27
57	15 + 1	3	21	7 + 1	118	25	3	42	17	179	32 + 1	4	91	20 + 1	240	39	4	124	27
58	15 + 1	3	22	7 + 1	119	25	3	43	17	180	32 + 1	4	92	20 + 1	241	39	4	125	27
59	16	3	19	8	120	25	3	44	17	181	32 + 1	4	93	20 + 1	242	39	4	126	27
60	16	3	20	8	121	25	3	45	17	182	33	4	90	21	243	39	4	127	27
61	16	3	21	8	122	25	3	46	17	183	33	4	91	21	244	40	4	124	28
									n = 245–488										
245	40	4	125	28	306	46	4	162	34	367	52	4	199	40	428	57	5	256	41
246	40	4	126	28	307	46	4	163	34	368	52	4	200	40	429	57	5	257	41
247	40	4	127	28	308	46	4	164	34	369	52	4	201	40	430	57	5	258	41
248	40	4	128	28	309	46	4	165	34	370	52	4	202	40	431	58	5	255	42
249	40	4	129	28	310	46	4	166	34	371	52	4	203	40	432	58	5	256	42
250	40	4	130	28	311	46	4	167	34	372	52	4	204	40	433	58	5	257	42
251	40	4	131	28	312	47	4	164	35	373	52	4	205	40	434	58	5	258	42
252	40	4	132	28	313	47	4	165	35	374	52	4	206	40	435	58	5	259	42
253	41	4	129	29	314	47	4	166	35	375	53	4	203	41	436	58	5	260	42
254	41	4	130	29	315	47	4	167	35	376	53	4	204	41	437	58	5	261	42
255	41	4	131	29	316	47	4	168	35	377	53	4	205	41	438	58	5	262	42
256	41	4	132	29	317	47	4	169	35	378	53	4	206	41	439	58	5	263	42
257	41	4	133	29	318	47	4	170	35	379	53	4	207	41	440	58	5	264	42
258	41	4	134	29	319	47	4	171	35	380	53	4	208	41	441	58	5	265	42
259	41	4	135	29	320	47	4	172	35	381	53	4	209	41	442	59	5	262	43
260	41	4	136	29	321	47	4	173	35	382	53	4	210	41	443	59	5	263	43
261	41	4	137	29	322	47	4	174	35	383	53 + 1	5	227	37 + 1	444	59	5	264	43
262	41	4	138	29	323	48	4	171	36	384	53 + 1	5	228	37 + 1	445	59	5	265	43
263	42	4	135	30	324	48	4	172	36	385	53 + 1	5	229	37 + 1	446	59	5	266	43
264	42	4	136	30	325	48	4	173	36	386	54	5	226	38	447	59	5	267	43
265	42	4	137	30	326	48	4	174	36	387	54	5	227	38	448	59	5	268	43
266	42	4	138	30	327	48	4	175	36	388	54	5	228	38	449	59	5	269	43
267	42	4	139	30	328	48	4	176	36	389	54	5	229	38	450	59	5	270	43
268	42	4	140	30	329	48	4	177	36	390	54	5	230	38	451	59	5	271	43
269	42	4	141	30	330	48	4	178	36	391	54	5	231	38	452	59	5	272	43
270	42	4	142	30	331	48	4	179	36	392	54	5	232	38	453	59	5	273	43
271	42	4	143	30	332	48	4	180	36	393	54	5	233	38	454	60	5	270	44
272	42	4	144	30	333	49	4	177	37	394	54	5	234	38	455	60	5	271	44
273	43	4	141	31	334	49	4	178	37	395	54 + 1	5	235	38 + 1	456	60	5	272	44
274	43	4	142	31	335	49	4	179	37	396	54 + 1	5	236	38 + 1	457	60	5	273	44
275	43	4	143	31	336	49	4	180	37	397	55	5	233	39	458	60	5	274	44
276	43	4	144	31	337	49	4	181	37	398	55	5	234	39	459	60	5	275	44
277	43	4	145	31	338	49	4	182	37	399	55	5	235	39	460	60	5	276	44
278	43	4	146	31	339	49	4	183	37	400	55	5	236	39	461	60	5	277	44
279	43	4	147	31	340	49	4	184	37	401	55	5	237	39	462	60	5	278	44
280	43	4	148	31	341	49	4	185	37	402	55	5	238	39	463	60	5	279	44
281	43	4	149	31	342	49	4	186	37	403	55	5	239	39	464	60	5	280	44
282	44	4	146	32	343	49	4	187	37	404	55	5	240	39	465	61	5	277	45
283	44	4	147	32	344	50	4	184	38	405	55	5	241	39	466	61	5	278	45
284	44	4	148	32	345	50	4	185	38	406	55	5	242	39	467	61	5	279	45
285	44	4	149	32	346	50	4	186	38	407	55	5	243	39	468	61	5	280	45
286	44	4	150	32	347	50	4	187	38	408	56	5	240	40	469	61	5	281	45
287	44	4	151	32	348	50	4	188	38	409	56	5	241	40	470	61	5	282	45

Table I continues

TABLE I. Continued

n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$
n = 245–488 (Continued)																			
288	44	4	152	32	349	50	4	189	38	410	56	5	242	40	471	61	5	283	45
289	44	4	153	32	350	50	4	190	38	411	56	5	243	40	472	61	5	284	45
290	44	4	154	32	351	50	4	191	38	412	56	5	244	40	473	61	5	285	45
291	44	4	155	32	352	50	4	192	38	413	56	5	245	40	474	61	5	286	45
292	45	4	152	33	353	50	4	193	38	414	56	5	246	40	475	61	5	287	45
293	45	4	153	33	354	51	4	190	39	415	56	5	247	40	476	61	5	288	45
294	45	4	154	33	355	51	4	191	39	416	56	5	248	40	477	62	5	285	46
295	45	4	155	33	356	51	4	192	39	417	56	5	249	40	478	62	5	286	46
296	45	4	156	33	357	51	4	193	39	418	56	5	250	40	479	62	5	287	46
297	45	4	157	33	358	51	4	194	39	419	56	5	251	40	480	62	5	288	46
298	45	4	158	33	359	51	4	195	39	420	57	5	248	41	481	62	5	289	46
299	45	4	159	33	360	51	4	196	39	421	57	5	249	41	482	62	5	290	46
300	45	4	160	33	361	51	4	197	39	422	57	5	250	41	483	62	5	291	46
301	45	4	161	33	362	51	4	198	39	423	57	5	251	41	484	62	5	292	46
302	46	4	158	34	363	51	4	199	39	424	57	5	252	41	485	62	5	293	46
303	46	4	159	34	364	51	4	200	39	425	57	5	253	41	486	62	5	294	46
304	46	4	160	34	365	52	4	197	40	426	57	5	254	41	487	62	5	295	46
305	46	4	161	34	366	52	4	198	40	427	57	5	255	41	488	62	5	296	46
n = 489–732																			
489	63	5	293	47	550	68	5	334	52	611	73	5	375	57	672	77	5	420	61
490	63	5	294	47	551	68	5	335	52	612	73	5	376	57	673	77	5	421	61
491	63	5	295	47	552	68	5	336	52	613	73	5	377	57	674	77	5	422	61
492	63	5	296	47	553	68	5	337	52	614	73	5	378	57	675	77	5	423	61
493	63	5	297	47	554	68	5	338	52	615	73	5	379	57	676	78	5	420	62
494	63	5	298	47	555	68	5	339	52	616	73	5	380	57	677	78	5	421	62
495	63	5	299	47	556	68	5	340	52	617	73	5	381	57	678	78	5	422	62
496	63	5	300	47	557	68	5	341	52	618	73	5	382	57	679	78	5	423	62
497	63	5	301	47	558	68	5	342	52	619	73	5	383	57	680	78	5	424	62
498	63	5	302	47	559	68	5	343	52	620	73	5	384	57	681	78	5	425	62
499	63	5	303	47	560	68	5	344	52	621	73	5	385	57	682	78	5	426	62
500	63	5	304	47	561	69	5	341	53	622	73	5	386	57	683	78	5	427	62
501	64	5	301	48	562	69	5	342	53	623	73	5	387	57	684	78	5	428	62
502	64	5	302	48	563	69	5	343	53	624	74	5	384	58	685	78	5	429	62
503	64	5	303	48	564	69	5	344	53	625	74	5	385	58	686	78	5	430	62
504	64	5	304	48	565	69	5	345	53	626	74	5	386	58	687	78	5	431	62
505	64	5	305	48	566	69	5	346	53	627	74	5	387	58	688	78	5	432	62
506	64	5	306	48	567	69	5	347	53	628	74	5	388	58	689	79	5	429	63
507	64	5	307	48	568	69	5	348	53	629	74	5	389	58	690	79	5	430	63
508	64	5	308	48	569	69	5	349	53	630	74	5	390	58	691	79	5	431	63
509	64	5	309	48	570	69	5	350	53	631	74	5	391	58	692	79	5	432	63
510	64	5	310	48	571	69	5	351	53	632	74	5	392	58	693	79	5	433	63
511	64	5	311	48	572	69	5	352	53	633	74	5	393	58	694	79	5	434	63
512	64	5	312	48	573	69	5	353	53	634	74	5	394	58	695	79	5	435	63
513	65	5	309	49	574	70	5	350	54	635	74	5	395	58	696	79	5	436	63
514	65	5	310	49	575	70	5	351	54	636	74	5	396	58	697	79	5	437	63
515	65	5	311	49	576	70	5	352	54	637	75	5	393	59	698	79	5	438	63
516	65	5	312	49	577	70	5	353	54	638	75	5	394	59	699	79	5	439	63
517	65	5	313	49	578	70	5	354	54	639	75	5	395	59	700	79	5	440	63
518	65	5	314	49	579	70	5	355	54	640	75	5	396	59	701	79	5	441	63
519	65	5	315	49	580	70	5	356	54	641	75	5	397	59	702	79	5	442	63
520	65	5	316	49	581	70	5	357	54	642	75	5	398	59	703	80	5	439	64
521	65	5	317	49	582	70	5	358	54	643	75	5	399	59	704	80	5	440	64

Table I continues

TABLE I. Continued

n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$	n	t	m	$\hat{n}$	$\hat{t}$
									n = 489–732 (Continued)										
522	65	5	318	49	583	70	5	359	54	644	75	5	400	59	705	80	5	441	64
523	65	5	319	49	584	70	5	360	54	645	75	5	401	59	706	80	5	442	64
524	65	5	320	49	585	70	5	361	54	646	75	5	402	59	707	80	5	443	64
525	66	5	317	50	586	71	5	358	55	647	75	5	403	59	708	80	5	444	64
526	66	5	318	50	587	71	5	359	55	648	75	5	404	59	709	80	5	445	64
527	66	5	319	50	588	71	5	360	55	649	75	5	405	59	710	80	5	446	64
528	66	5	320	50	589	71	5	361	55	650	76	5	402	60	711	80	5	447	64
529	66	5	321	50	590	71	5	362	55	651	76	5	403	60	712	80	5	448	64
530	66	5	322	50	591	71	5	363	55	652	76	5	404	60	713	80	5	449	64
531	66	5	323	50	592	71	5	364	55	653	76	5	405	60	714	80	5	450	64
532	66	5	324	50	593	71	5	365	55	654	76	5	406	60	715	80	5	451	64
533	66	5	325	50	594	71	5	366	55	655	76	5	407	60	716	81	5	448	65
534	66	5	326	50	595	71	5	367	55	656	76	5	408	60	717	81	5	449	65
535	66	5	327	50	596	71	5	368	55	657	76	5	409	60	718	81	5	450	65
536	66	5	328	50	597	71	5	369	55	658	76	5	410	60	719	81	5	451	65
537	67	5	325	51	598	71	5	370	55	659	76	5	411	60	720	81	5	452	65
538	67	5	326	51	599	72	5	367	56	660	76	5	412	60	721	81	5	453	65
539	67	5	327	51	600	72	5	368	56	661	76	5	413	60	722	81	5	454	65
540	67	5	328	51	601	72	5	369	56	662	76	5	414	60	723	81	5	455	65
541	67	5	329	51	602	72	5	370	56	663	77	5	411	61	724	81	5	456	65
542	67	5	330	51	603	72	5	371	56	664	77	5	412	61	725	81	5	457	65
543	67	5	331	51	604	72	5	372	56	665	77	5	413	61	726	81	5	458	65
544	67	5	332	51	605	72	5	373	56	666	77	5	414	61	727	81	5	459	65
545	67	5	333	51	606	72	5	374	56	667	77	5	415	61	728	81	5	460	65
546	67	5	334	51	607	72	5	375	56	668	77	5	416	61	729	81	5	461	65
547	67	5	335	51	608	72	5	376	56	669	77	5	417	61	730	82	5	458	66
548	67	5	336	51	609	72	5	377	56	670	77	5	418	61	731	82	5	459	66
549	68	5	333	52	610	72	5	378	56	671	77	5	419	61	732	82	5	460	66

The calculation of $\lceil n^{2/3} \rceil$ for M_n is only calculated once as specified in step (2). Furthermore, based on the above observations, 2 is only added at most once to $\lceil n^{2/3} \rceil$ to determine a feasible transmitting time of M_n, i.e., in the later reduction steps, the transmitting times are always feasible for the reduced meshes. We also note that Q can be partitioned into two sets A and $Q - A$ according to Table I, where

$$A = \{47{:}49,\ 55{:}58,\ 63,\ 64,\ 149{:}152,\ 156{:}162,\ 167{:}172,\ 178{:}181,\ 189,\ 383{:}385,\ 395,\ 396\}.$$

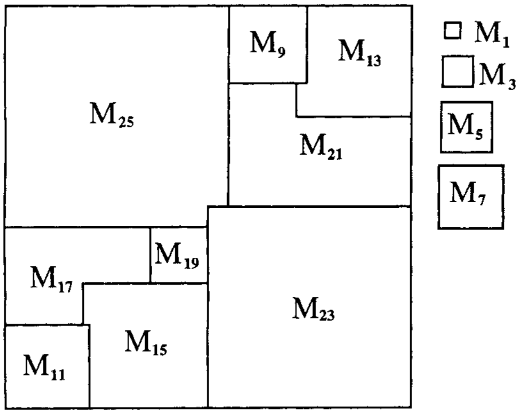

Fig. 4. A manual solution for M_{45}.

We choose $t = \lceil n^{2/3} \rceil + 1$ for $n \in A$, and $t = \lceil n^{2/3} \rceil + 2$ for $n \in Q - A$. Thus, step (6) in Algorithm $TA(n, t)$ is replaced as follows (initial value of *flag* is 0):

(6) If *flag* = 0, set t to be $t + 1$ for $n \in A$, and $t + 2$ for $n \in Q - A$; set *flag* = 1.

Hence, we can conclude that the proposed algorithm gives a transmitting scheme with $t = \lceil n^{2/3} \rceil + 1$ for $n \in A \cup \{8, 11\}$, $t = \lceil n^{2/3} \rceil + 2$ for $n \in Q - A$, and $t = \lceil n^{2/3} \rceil$ otherwise. Thus, this transmitting problem is solved by the proposed transmitting scheme, though not optimal. We give the performance bound in the following theorem:

Theorem 5. *The ratio of the proposed transmitting time to the optimal transmitting time is approximate to* 1.1.

Proof. Algorithm $TA(n, t)$ yields a feasible transmitting time t of $\lceil n^{2/3}\rceil + 1$, $\lceil n^{2/3}\rceil + 2$, or $\lceil n^{2/3}\rceil$, depending on n. Let t^{opt} denote the optimal transmitting time for M_n and t^* be the smallest integer satisfying $\frac{4}{3}t^{*3} - \frac{1}{3}t^* \geq n^2$. Obviously, we have $t^{opt} \geq t^*$. When n is given, we have $t^* \geq t_1$, where $t_1 = \lceil\sqrt[3]{\frac{3}{4}}n^{2/3}\rceil$, the smallest integer satisfying $\frac{4}{3}t_1^3 \geq n^2$. Thus, the ratio of the proposed transmitting time to the optimal transmitting time is given as follows:

$$\frac{t}{t^{opt}} \leq \frac{t}{t^*} \leq \frac{\lceil n^{2/3}\rceil + 2}{\lceil\sqrt[3]{\frac{3}{4}}n^{2/3}\rceil} \sim \sqrt[3]{\frac{4}{3}} \sim 1.1.$$

Hence, the theorem follows. ■

6. TRANSMITTING ON DE BRUIJN GRAPHS

The (directed) de Bruijn graph B_d^n, called a d-ary n-dimensional de Bruijn graph, is a directed graph having the vertex set $V(B_d^n) = \{0, 1, \ldots, d^n - 1\}$. Each vertex v in B_d^n can be expressed as $v = (v_{n-1}, v_{n-2}, \ldots, v_0)$, where $0 \leq v_i \leq d - 1$ for all $0 \leq i \leq n - 1$, and $v = \sum_{i=0}^{n} v_i d^i$. Each vertex is denoted by its label v or its n-tuple representation, which are used interchangeably. Each vertex v is connected to vertex u, denoted by (v, u), where $u = (v_{n-2}, v_{n-3}, \ldots, v_0, \alpha)$ and $0 \leq \alpha \leq d - 1$. By convention, when $n = 1$, B_d^1 is a complete directed graph on d vertices with self-loops. Each vertex of B_d^n has outdegree and indegree d.

In this section, we give the optimal transmitting cost for B_d^n, which is given in the following theorem, and the optimal transmitting schemes can be found in the proof of the theorem.

Theorem 6. *The optimal transmitting cost of* B_d^n *is* $(1, 1)$ *when* $d = 1$, $(2, 1)$ *when* $n = 1$, $(n, 2)$ *when* $d = 2$, *and* $(n + 1, 1)$ *when* $d \geq 3$.

Proof. It is trivial to verify the cases of $d = 1$ or $n = 1$. Now, we consider the cases that $d \geq 2$ and $n \geq 2$. Since B_d^n has outdegree d, it follows that for any vertex v and a nonnegative integer $r \leq n$ we have $|N_r(v)| \leq 1 + d + d^2 + \cdots + d^r = (d^{r+1} - 1)/(d - 1)$. Hence,

$$|\cup_{f(i)\in V} N_{t-i}(f(i))| \leq \sum_{r=0}^{t-1} \frac{d^{r+1} - 1}{d - 1} = \frac{d(d^t - 1)}{(d-1)^2} - \frac{t}{d-1}. \tag{9}$$

Consider $d \geq 3$. Since for any vertex v we have $N_n(v) = V(B_d^n)$, we can give an $(n + 1, 1)$-transmitting scheme as follows: The host sends a message to an arbitrary vertex in B_d^n and then idles. After $n + 1$ units of time, all processors in B_d^n can receive the message. Suppose that the optimal transmitting time is $t \leq n$. It follows that

$$|\cup_{f(i)\in V} N_{t-i}(f(i))| \leq \frac{d(d^n - 1)}{(d-1)^2} - \frac{n}{d-1} < d^n.$$

It means that not all the vertices in B_d^n can receive the message in n time units, which is a contradiction. Thus, the $(n + 1, 1)$-transmitting scheme is optimal for B_d^n when $d \geq 3$.

Consider $d = 2$. Let (t^*, s^*) be the optimal transmitting cost for B_2^n. For all (t, s)-transmitting schemes, it follows from (9) that $|\cup_{f(i)\in V} N_{t-i}(f(i))| \leq 2^n - n - 1 < 2^n$ if $t \leq n - 1$. Thus, we have $t \geq n$, and, in particular, $t^* \geq n$. Suppose that $t^* = n$ and $s^* = 1$. It follows that $|\cup_{f(i)\in V} N_{t-i}(f(i))| = |N_{n-1}(f(1))| \leq \sum_{i=1}^{n} 2^{i-1} = 2^n - 1 < 2^n$, which is a contradiction. Thus, if $t^* = n$, we must have $s^* \geq 2$. We give an $(n, 2)$-transmitting scheme f as follows: $f(1) = 1$, $f(k) = 0$ for an arbitrary k satisfying $2 \leq k \leq n$, and $f(i) = v_0$ for all $i \neq 1, k$. In other words, the host sends a message to vertex 1, i.e., $(0, \ldots, 0, 1)$, at the first time unit, and then to vertex 0, i.e., $(0, \ldots, 0)$ at the k-th unit of time, $2 \leq k \leq n$. For any vertex $v = (v_{n-1}, v_{n-2}, \ldots, v_0)$, $2 \leq v \leq 2^n - 1$, it is obvious that $d(1, v) \leq n - 1$ since $v_i = 1$ for some $1 \leq i \leq n - 1$. Therefore, $N_{n-1}(1) = V(B_d^n) - \{0\}$. Thus, $N_{n-1}(1) \cup f(k) = N_{n-1}(1) \cup \{0\} = V(B_d^n)$. Hence, the proposed $(n, 2)$-transmitting scheme is optimal. The theorem follows. ■

7. CONCLUSION

Linear arrays and (token) rings are of practical use in networks. Binary trees are also a useful topology for parallel systems. We present optimal transmitting schemes on rings, linear arrays, complete binary trees, and star trees. In observing the solution method for star trees presented in Section 4, we need more number theory results to solve optimal transmitting problems on starlike graphs. Not surprisingly, solving the optimal transmitting scheme for general trees is much more difficult.

In transmitting on diagonal meshes, we can only give an approximation solution, not an optimal one. The proof for showing that the proposed function f is a transmitting scheme has already involved very complicated calculations. It is even more difficult to give an optimal transmitting scheme on (conventional) meshes or tori.

In the above discussion, the host is first assumed to have a link to every processor in the graph. However, only $s(G)$ links between the host processor and the processors in G are sufficient for transmitting purposes.

The authors would like to thank Dr. Frank Hsu of Fordham University for his helpful suggestions and discussions in preparing this paper. This paper was partially supported by the National Science Council, Republic of China.

REFERENCES

[1] N. Alon, Transmitting in the n-dimensional cube. *Discr. Appl. Math.* **37/38** (1992) 9–11.

[2] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications.* Elsevier, New York (1976).

[3] N. G. de Bruijn, A combinatorial problem. *Nederl. Akad. Wetensch. Proc. Ser. A* **49** (1946) 758–764.

[4] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes.* Morgan Kaufmann, San Mateo, CA (1992).

[5] Y. Yang and H. Amano, Message transfer algorithms on the recursive diagonal torus. *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks* (1994) 310–317.

[6] Y. Yang, H. Amano, H. Shibamura, and T. Sueyoshi, Recursive diagonal torus: An interconnection network for massively parallel computers. *Proceedings of the 5th IEEE Symposium on Parallel and Distributed Processing* (1993) 591–594.

Received September 13, 1993
Accepted September 5, 1995