Journal of Parallel and Distributed Computing **57**, 271–294 (1999)
Article ID jpdc.1999.1538, available online at http://www.idealibrary.com on IDEAL®

Broadcasting on Incomplete WK-Recursive Networks

Ming-Yang Su

Department of Electronic Engineering, Hwa Hsia College of Technology and Commerce, Taipei, Taiwan

Gen-Huey Chen*

Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
E-mail: ghchen@csie.ntu.edu.tw

and

Dyi-Rong Duh

Department of Computer Science and Information Engineering, National Chi Nan University, Taipei, Taiwan

Received January 5, 1996; revised January 29, 1999; accepted February 1, 1999

The WK-recursive networks, which were originally proposed by Vecchia and Sanges, have suffered from a rigorous restriction on the number of nodes. Like other incomplete networks, the incomplete WK-recursive networks have been proposed to relieve this restriction. In this paper, broadcasting on the incomplete WK-recursive networks is discussed. The proposed broadcasting algorithm is optimal with respect to message complexity. Besides, extensive experiments are made to evaluate its performance. Experimental results show that (1) the heights of the broadcasting trees do not exceed the diameters, (2) a high percentage of the nodes can receive the message from the source node via the shortest paths, (3) for those nonshortest transmission paths, the deviations are small, and (4) a high percentage of the broadcasting trees are of minimum height.

Key Words: broadcasting algorithms; graph-theoretic interconnection networks; incomplete WK-recursive networks; WK-recursive networks.

* Corresponding author.

0743-7315/99 $30.00

1. INTRODUCTION

One crucial step on designing a large-scale multiprocessor system is to determine the topology of the interconnection network (network for short). In the recent decade, a number of networks have been announced in the literature [1, 5, 13, 24, 26, 29, 34]. Among them, the WK-recursive networks own two structural advantages: expansibility and equal degree. A network is *expansible* if no changes to node configuration and link connection are necessary when it is expanded, and of *equal degree* if all its nodes have the same degree no matter what the size is. A network with these two properties will gain the advantages of easy implementation and low cost when it is manufactured.

The WK-recursive networks, which were originally proposed by Vecchia and Sanges [34], represent a family of recursively scalable networks. They offer a high degree of regularity, scalability, and symmetry which very well conform to a modular design and implementation of distributed systems involving a large number of computing elements. A VLSI implementation of a 16-node WK-recursive network was realized at the Hybrid Computing Research Center [34]. In this implementation, each node was implemented with the INMOS IMS T414 transputer, and the network processes were coded in Occam programming language. Later this prototype network was further extended to 64 nodes [35]. Recently two variants of the WK-recursive networks have been proposed in [8, 9].

Much research [4, 6, 7, 10–12, 14, 34, 35] was devoted to the WK-recursive networks. Vecchia and Sanges first presented a heuristic routing algorithm [34] and an optimal broadcasting algorithm [35]. Chen and Duh [4] then improved their work by presenting a shortest-path routing algorithm and another optimal, but simpler, broadcasting algorithm. Topological properties such as diameter, connectivity, and hamiltonicity were also investigated in [4]. Parallel routing paths, wide diameter, and fault diameter were computed in [6]. The Rabin number problem was discussed in [7]. Edge-disjoint hamiltonian paths and edge-disjoint spanning trees were constructed in [11]. Some substructure allocation algorithms for a multiuser WK-recursive network were presented in [10]. Embedding rings in a faulty WK-recursive network appeared in [12]. A special multicast wormhole routing was discussed in [14].

Although the WK-recursive networks own many favorable properties, there is a rigorous restriction on the number of their nodes. As will become clear in the next section, the number of nodes in a WK-recursive network must satisfy d^t, where $d > 1$ is the size of the basic building block and $t \geqslant 1$ is the level of expansion. Thus, as $d = 4$, extra $3 \times 4^7 = 49{,}152$ nodes are required to expand from a 7-level WK-recursive network to an 8-level one. Almost all the announced networks have suffered from the same restriction. To relieve this restriction, some incomplete networks [17–19, 23, 25, 30] have been defined recently. Katseff [17] defined the incomplete hypercubes that allow the number of nodes to be any positive integer. Latifi and Bagherzadeh [18], and Ravikumar, Kuchlous, and Manimaran [25] independently defined the incomplete star networks that contain an arbitrary number of nodes. Later Latifi and Bagherzadeh [19] defined a special class of the incomplete star networks, named the *clustered-star graphs*, which contain $c \cdot k!$

nodes, where $1 < c \leqslant k$ is a positive integer. Ponnuswamy and Chaudhary [23] defined the incomplete rotator graphs that have the same restriction as the clustered-star graphs in the number of nodes. Recently, the incomplete WK-recursive networks have been defined by the authors [30]. With practical consideration, the number of nodes is required to be a multiple of d, where d is the size of the basic building block. Thus an incomplete WK-recursive network can be expanded or contracted in units of basic building blocks. Besides these incomplete networks, some enhanced incomplete networks have also been proposed in the literature [27, 33].

In order for an incomplete network to be considered a general-purpose parallel and distributed system, efficient communication must be addressed because communication delay has been recognized as a major cause of performance degradation in such a computing environment (see [2, 15]). Undoubtedly, broadcasting is one of the most fundamental problems in the study of communication within a network, and so it is not surprising that much research [16, 21, 22, 28, 35] has been devoted to the subject of broadcasting. In [17], Katseff presented an optimal broadcasting algorithm for the incomplete hypercubes. In [18], Latifi and Bagherzadeh presented a broadcasting algorithm for a special class of the incomplete star networks that comprise c k-stars, where $1 < c \leqslant k$ (i.e., the clustered-star graphs as defined in [19]). To the best of our knowledge, no broadcasting algorithms for the (general) incomplete star networks and the incomplete rotator graphs have been suggested.

In this paper, a broadcasting algorithm for the incomplete WK-recursive networks is proposed. Since the structures of the incomplete WK-recursive networks can be expressed as multistage graphs, the broadcasting problem is reduced to constructing spanning trees for the representative multistage graphs. The proposed broadcasting algorithm has optimal message complexity. Besides, extensive experiments were made to verify its efficiency. The rest of this paper is organized as follows. In the next section, the incomplete WK-recursive networks are formally defined and their multistage graph representations are introduced. Broadcastings for two special situations are first described in Sections 3 and 4, respectively. Then, broadcasting for a general situation is described in Section 5. The performance of the broadcasting algorithm is evaluated in Section 6 with extensive experiments. Finally, this paper is concluded in Section 7.

2. INCOMPLETE WK-RECURSIVE NETWORKS AND THEIR MULTISTAGE GRAPH REPRESENTATIONS

In this section we first review the structures of the WK-recursive networks. Some notations and definitions related to the WK-recursive networks are also introduced. In terms of graph [3], the incomplete WK-recursive networks are induced subgraphs of the WK-recursive networks. In this paper, graphs and networks are used interchangeably. The diameters, connectivities, and hamiltonian circuits of the incomplete WK-recursive networks were computed in [31]. A shortest-path routing algorithm for the incomplete WK-recursive networks was presented in [30].

The WK-recursive networks can be constructed recursively by grouping basic building blocks. Any complete graph can serve as a basic building block. For convenience, let $K(d, t)$ denote a WK-recursive network of level t whose basic building blocks are each a d-node complete graph, where $d > 1$ and $t \geqslant 1$. $K(d, 1)$, which is the basic building block, is a d-node complete graph, and $K(d, t)$ for $t \geqslant 2$ can be constructed by connecting d $K(d, t-1)$'s as a d-supernode complete graph (each $K(d, t-1)$ is regarded as a supernode). Each node of $K(d, t)$ is associated with a t-digit identifier. The following definition is due to Chen and Duh [4].

DEFINITION 2.1. The node set of $K(d, t)$ is denoted by $\{a_{t-1}a_{t-2}\cdots a_1a_0 \mid a_i \in \{0, 1, ..., d-1\}$ for $0 \leqslant i \leqslant t-1\}$. Node adjacency is defined as follows: $a_{t-1}a_{t-2}\cdots a_1a_0$ is adjacent to (1) $a_{t-1}a_{t-2}\cdots a_1b$, where $0 \leqslant b \leqslant d-1$ and $b \neq a_0$, and (2) $a_{t-1}a_{t-2}\cdots a_{j+1}a_{j-1}(a_j)^j$ if $a_j \neq a_{j-1}$ and $a_{j-1} = a_{j-2} = \cdots = a_1 = a_0$, where $1 \leqslant j \leqslant t-1$ and $(a_j)^j$ represents j consecutive a_j's. The links of (1), which are labeled 0, are called *substituting links*. The links of (2), which are labeled j, are called *j-flipping links* (or simply *flipping links*). Besides, there are *open links* whose one end node is $(a)^t$, where $0 \leqslant a \leqslant d-1$, and the other end node is unspecified. The open links are labeled t.

$K(d, t)$ contains d^t nodes. Since each node is incident with $d-1$ substituting links and one flipping link (or open link), $K(d, t)$ has degree d. The structures of $K(4, 1)$ and $K(4, 3)$ are illustrated in Fig. 1. The substituting links are within basic building blocks, whereas each j-flipping link connects two embedded $K(d, j)$'s. The open links are left for future expansion. For example, let us consider the incident links of node 311 in Fig. 1. The one to node 133 is a 2-flipping link; the others are substituting links.

DEFINITION 2.2. Define $c_{t-1}c_{t-2}\cdots c_m \cdot K(d, m)$ to be the induced subgraph of $K(d, t)$ by $\{c_{t-1}c_{t-2}\cdots c_m a_{m-1}\cdots a_1a_0 \mid a_j \in \{0, 1, ..., d-1\}$ for $0 \leqslant j \leqslant m-1\}$, where $1 \leqslant m < t$ and $c_{t-1}, c_{t-2}, ..., c_m$ are all integers from $\{0, 1, ..., d-1\}$. That is, $c_{t-1}c_{t-2}\cdots c_m \cdot K(d, m)$ is an embedded $K(d, m)$ with identifier $c_{t-1}c_{t-2}\cdots c_m$. For example (refer to Fig. 1), $31 \cdot K(4, 1)$ is the induced subgraph of $K(4, 3)$ by $\{310, 311, 312, 313\}$.

DEFINITION 2.3. Node $a_{t-1}a_{t-2}\cdots a_1a_0$ is a *k-frontier* if $a_{k-1} = \cdots = a_1 = a_0$, where $1 \leqslant k \leqslant t$.

By Definition 2.3, a k-frontier is automatically an l-frontier for $1 \leqslant l < k$. Both end nodes of a k-flipping link are k-frontiers. An embedded $K(d, m)$ contains one $(m+1)$-frontier and $d-1$ m-frontiers. These d frontiers are $2^m - 1$ distant from each other.

The incomplete WK-recursive networks, which were originally defined in [30], are induced subgraphs of the WK-recursive networks. If we number the nodes of $K(d, t)$ according to their lexicographical order, then an N-node incomplete WK-recursive network is the subgraph of $K(d, t)$ induced by the first N nodes. Throughout this paper, we use $IK(d, t)$ to denote an N-node incomplete WK-recursive network, where $d^{t-1} < N < d^t$ and N is a multiple of d.

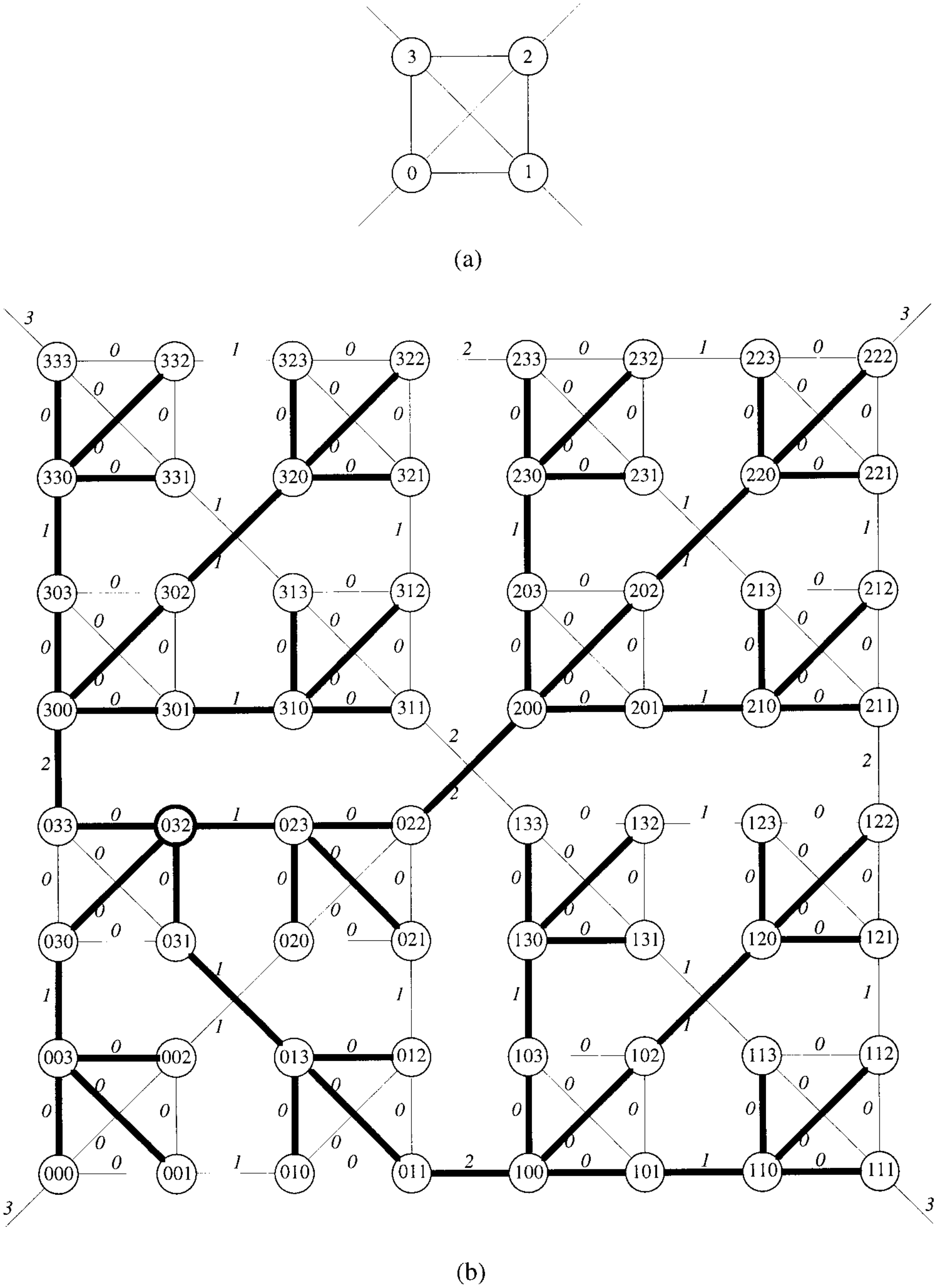


FIG. 1. The structures of (a) K(4, 1) and (b) K(4, 3). Figure 1b also shows the spanning tree rooted at 032 that results from Chen and Duh's broadcasting algorithm.

The *coefficient vector* of IK(d, t) is uniquely defined as a $(t-1)$-vector $(b_{t-1}, b_{t-2}, ..., b_1)$ such that $N = b_{t-1}d^{t-1} + b_{t-2}d^{t-2} + \cdots + b_1 d$. IK$(d, t)$ with coefficient vector $(b_{t-1}, b_{t-2}, ..., b_1)$ contains b_m embedded K(d, m)'s with identifiers $b_{t-1}b_{t-2}\cdots b_{m+1}0$, $b_{t-1}b_{t-2}\cdots b_{m+1}1$, ..., and $b_{t-1}b_{t-2}\cdots b_{m+1}(b_m - 1)$, respectively, where $1 \leqslant m \leqslant t-1$. For example, IK(5, 6) with coefficient vector (2, 3, 0, 4, 2) contains the following embedded $\mathrm{K}(d, m)$'s:

$$0 \cdot \mathrm{K}(5,5),\ 1 \cdot \mathrm{K}(5,5)$$

$$20 \cdot \mathrm{K}(5,4),\ 21 \cdot \mathrm{K}(5,4),\ 22 \cdot \mathrm{K}(5,4)$$

$$2300 \cdot \mathrm{K}(5,2),\ 2301 \cdot \mathrm{K}(5,2),\ 2302 \cdot \mathrm{K}(5,2),\ 2303 \cdot \mathrm{K}(5,2)$$

$$23040 \cdot \mathrm{K}(5,1),\ 23041 \cdot \mathrm{K}(5,1)$$

Figure 2 shows the structure of IK(4, 3) with coefficient vector (3, 2). In the rest of this paper, a coefficient vector $(b_{t-1}, b_{t-2}, ..., b_1)$ is written as $(b_{t-1}, b_{t-2}, ..., b_i, *)$, where $1 \leqslant i \leqslant t-1$, provided $b_i \neq 0$ and $b_{i-1} = b_{i-2} = \cdots = b_1 = 0$. For example, (2, 3, 0, 4, 0) is written as (2, 3, 0, 4, $*$), and (2, 3, 4) is written as (2, 3, 4, $*$).

Let S_m represent the subgraph of $\mathrm{IK}(d, t)$ with coefficient vector $(b_{t-1}, b_{t-2}, ..., b_i, *)$ induced by the nodes of $b_{t-1} b_{t-2} \cdots b_{m+1} 0 \cdot \mathrm{K}(d, m)$, $b_{t-1} b_{t-2} \cdots b_{m+1} 1 \cdot \mathrm{K}(d, m)$, ..., and $b_{t-1} b_{t-2} \cdots b_{m+1} (b_m - 1) \cdot \mathrm{K}(d, m)$, where $i \leqslant m \leqslant t-1$. That is,

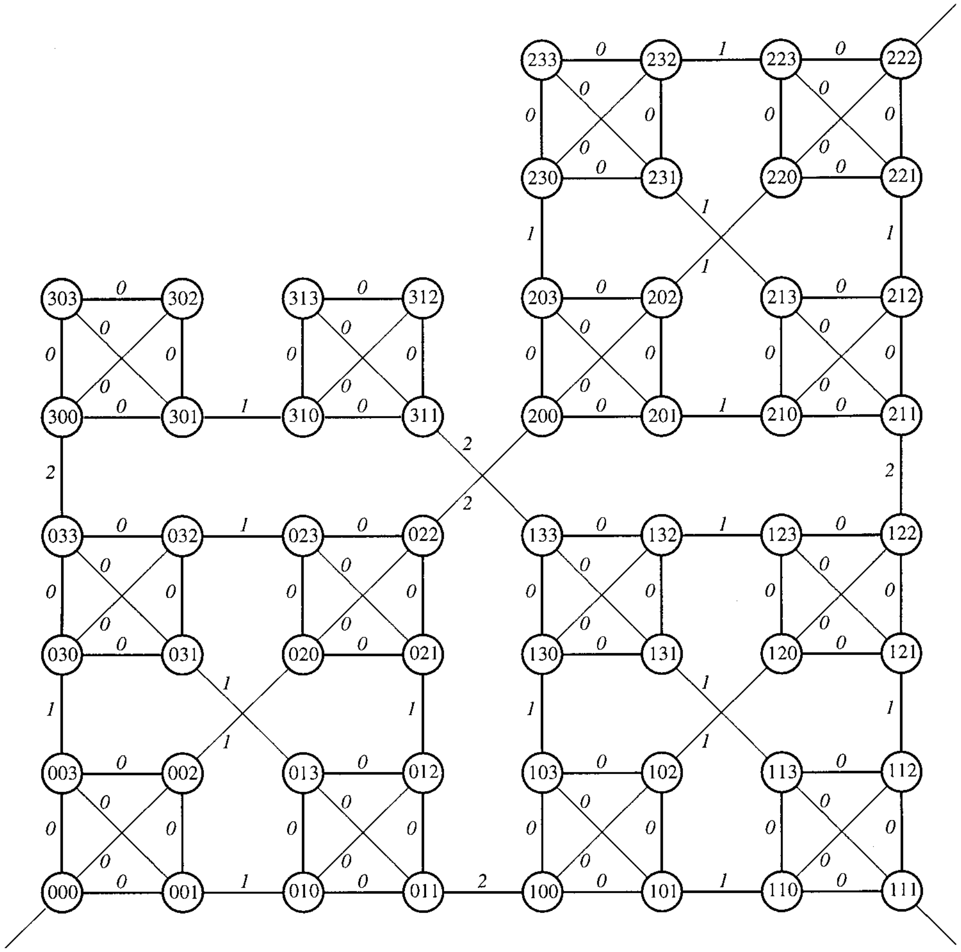


FIG. 2. The structure of IK(4, 3) with coefficient vector (3, 2).

S_m contains b_m embedded K(d, m)'s with identifiers $b_{t-1}b_{t-2}\cdots b_{m+1}0$, $b_{t-1}b_{t-2}\cdots b_{m+1}1$, ..., and $b_{t-1}b_{t-2}\cdots b_{m+1}(b_m - 1)$, respectively. We note that there is an m-flipping link between any two of these b_m embedded K(d, m)'s. For example, the IK(5, 6) with coefficient vector (2, 3, 0, 4, 2, $*$) mentioned above has S_5 containing $0 \cdot$ K(5, 5) and $1 \cdot$ K(5, 5), S_4 containing $20 \cdot$ K(5, 4), $21 \cdot$ K(5, 4), and $22 \cdot$ K(5, 4), S_3 empty, S_2 containing $2300 \cdot$ K(5, 2), $2301 \cdot$ K(5, 2), $2302 \cdot$ K(5, 2), and $2303 \cdot$ K(5, 2), and S_1 containing $23040 \cdot$ K(5, 1) and $23041 \cdot$ K(5, 1).

According to the discussion above, the structure of IK(d, t) with coefficient vector $(b_{t-1}, b_{t-2}, ..., b_i, *)$ can be expressed as a $(t-i)$-stage graph, regarding each S_m as a stage. The $(t-i)$-stage graph is denoted by $S_{t-1} + S_{t-2} + \cdots + S_i$. For example, IK(5, 6) with coefficient vector (2, 3, 0, 4, 2, $*$) can be expressed as a five-stage graph as shown in Fig. 3. For simplicity each embedded K(d, m) within S_m is drawn as a circle, and the one with identifier $b_{t-1}b_{t-2}\cdots b_{m+1}j$, where $0 \leqslant j \leqslant b_{m-1}$, is denoted by C^j_m. All the m-flipping links between these circles are omitted for conciseness. Also note that for $t-1 \geqslant m \geqslant n \geqslant i$, $S_m + S_{m-1} + \cdots + S_n$ itself forms an embedded IK$(d, m+1)$ with coefficient vector $(b_m, b_{m-1}, ..., b_n, *)$, in which each node has its identifier prefixed with $b_{t-1}b_{t-2}\cdots b_{m+1}$.

There are $\min\{b_m, b_{m-1}\}$ m-flipping links between S_m and S_{m-1}, each connecting C^j_m and C^j_{m-1} for some $0 \leqslant j \leqslant \min\{b_m, b_{m-1}\} - 1$. Besides, for $t-1 \geqslant u > v \geqslant i$ and $u - v > 1$, there may exist a u-flipping link between S_u and S_v. If such a link exists,

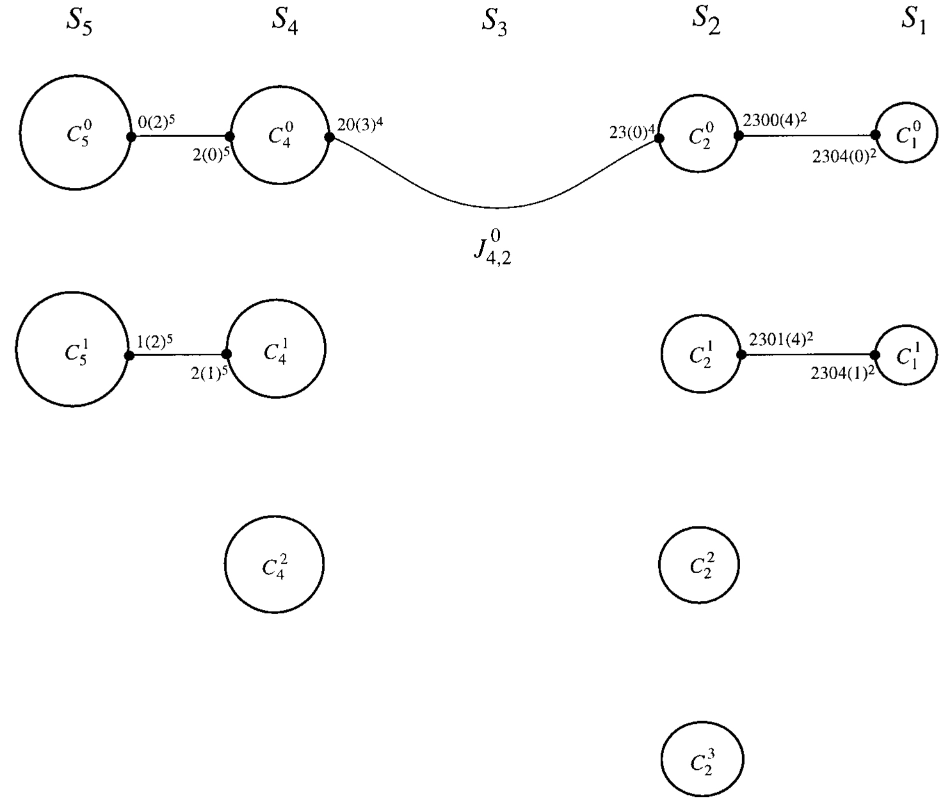


FIG. 3. The multistage graph representation of IK(6, 5) with coefficient vector (2, 3, 0, 4, 2).

it is called a *jumping u-flipping link*. A necessary and sufficient condition for the existence of jumping flipping links was suggested in [30] as follows.

THEOREM 2.1 [30]. *For* IK(d, t) *with coefficient vector* $(b_{t-1}, b_{t-2}, ..., b_i, *)$, *one jumping u-flipping link exists between* S_u *and* S_v *if and only if* $b_u > b_{u-1} = b_{u-2} = \cdots = b_{v+1} < b_v$, *where* $t-1 \geqslant u > v \geqslant i$ *and* $u - v > 1$. *Moreover, this jumping flipping link connects* C_u^e *and* C_v^e, *where* $e = b_{u-1} = b_{u-2} = \cdots = b_{v+1}$.

According to Theorem 2.1, there is a simple method to determine all jumping flipping links from the coefficient vector $(b_{t-1}, b_{t-2}, ..., b_i, *)$. We only need to examine $(b_{t-1}, b_{t-2}, ..., b_i, *)$ from the left to the right, and a jumping u-flipping link exists between C_u^e and C_v^e if $u - v > 1$ and $b_u > b_{u-1} = b_{u-2} = \cdots = b_{v+1} < b_v$, where $e = b_{u-1} = b_{u-2} = \cdots = b_{v+1}$. In the rest of this paper we use $J_{u,v}^e$ to denote the jumping u-flipping link between C_u^e and C_v^e (refer to Fig. 3 for illustration).

The structure of $S_{t-1} + S_{t-2} + \cdots + S_i$ is further detailed as follows. Since each C_m^j, where $t-1 \geqslant m \geqslant i$ and $0 \leqslant j \leqslant b_m - 1$, is a K$(d, m)$, the links inside C_m^j are subject to Definition 2.1. On the other hand, the links incident to C_m^j include (1) $b_m - 1$ m-flipping links connecting $C_m^0, C_m^1, ..., C_m^{j-1}, C_m^{j+1}, ...,$ and $C_m^{b_m - 1}$, respectively; (2) one m-flipping link connecting C_{m-1}^j if $j \leqslant b_{m-1} - 1$, or one jumping m-flipping link connecting C_l^j, where $l < m-1$, if $j = b_{m-1} = b_{m-2} = \cdots = b_{l+1} < b_l$; (3) one $(m+1)$-flipping link connecting C_{m+1}^j if $j \leqslant b_{m+1} - 1$, or one jumping h-flipping link connecting C_h^j, where $h > m+1$, if $b_h > b_{h-1} = b_{h-2} = \cdots = b_{m+1} = j$. Both end nodes of (1) are $b_{t-1} b_{t-2} \cdots b_{m+1} j(x)^m \in C_m^j$ and $b_{t-1} b_{t-2} \cdots b_{m+1} x(j)^m \in C_m^x$, where $0 \leqslant x \leqslant b_m - 1$ and $x \neq j$. Both end nodes of (2) are $b_{t-1} b_{t-2} \cdots b_{m+1} (b_m)^m \in C_m^j$ and $b_{t-1} b_{t-2} \cdots b_{m+1} b_m (j)^m \in C_{m-1}^j$ (or $\in C_l^j$). Both end nodes of (3) are $b_{t-1} b_{t-2} \cdots b_{m+2}\, j (b_{m+1})^{m+1} \in C_{m+1}^j$ (or $b_{t-1} b_{t-2} \cdots b_{h+1}\, j (b_h)^h \in C_h^j$) and $b_{t-1} b_{t-2} \cdots b_{m+2} b_{m+1} (j)^{m+1} \in C_m^j$.

Since the structure of IK(d, t) can be expressed as a multistage graph, message-optimal broadcasting in IK(d, t) is equivalent to constructing a spanning tree for the multistage graph. The spanning tree is also referred to as broadcasting tree when broadcasting is concerned. Suppose the source node is $r \in S_z$ and broadcasting is performed on IK(d, t) with coefficient vector $(b_{t-1}, b_{t-2}, ..., b_i, *)$, where $t-1 \geqslant z \geqslant i$. The resulting spanning tree is denoted by $ST(t-1, i, r, z)$. In subsequent sections, without loss of generality, we describe the broadcasting algorithm by explaining how to construct $ST(m, n, r, z)$ in $S_m + S_{m-1} + \cdots + S_n$, where $t-1 \geqslant m \geqslant z \geqslant n \geqslant i$. In the next two sections, we first construct $ST(m, n, r, z)$ for two special cases: $z = m$ and $z = n$. Then, constructing $ST(m, n, r, z)$ for arbitrary z ranging from n to m is discussed in Section 5.

3. CONSTRUCTING $ST(m, n, r, m)$

Several basic dissemination patterns lay the foundation of our construction algorithm for $ST(m, n, r, m)$. The purpose of this section is to introduce them and show how they can be used to construct $ST(m, n, r, m)$. First of all, we have to review Chen and Duh's broadcasting algorithm [4] for K(d, t) because the algorithm will be executed by the dissemination patterns. Chen and Duh's algorithm requires a stack of length $t+1$ (a bit array of length $t+1$ for real implementation), which keeps the labels of links, to be carried with the message. Initially, the source node pushes the label t into the stack and disseminates the message over all its incident links but the one with label t. Once a node receives the message via its one incident link with label, say k, it further disseminates the message by executing the steps:

1. Pop elements of the stack until the top element is greater than k.
2. Push k into the stack.
3. Disseminate the message over those incident links whose labels do not appear in the stack.

For illustration, Fig. 1(b) shows with bold lines the broadcasting tree that results from executing Chen and Duh's algorithm on K(4, 3) with source node 032 (that is, the spanning tree rooted at node 032). The following lemma was proved in [4].

LEMMA 3.1 [4]. *Starting from any node, Chen and Duh's algorithm can disseminate a message to each node of* $\mathrm{K}(d, t)$ *exactly once. Moreover, the resulting spanning tree has height at most* $2^t - 1$, *which is the diameter of* $\mathrm{K}(d, t)$.

With slight modification, Chen and Duh's algorithm can disseminate a message within any embedded $\mathrm{K}(d, l)$, where $1 \leqslant l < t$. We assume $r = r_{t-1} r_{t-2} \cdots r_1 r_0$. A spanning tree of $r_{t-1} r_{t-2} \cdots r_{l+1} r_l \cdot \mathrm{K}(d, l)$ rooted at node r can be obtained by executing Chen and Duh's algorithm, provided step 3 is modified as

3′. Disseminate the message over those incident links whose labels are smaller than l and do not appear in the stack.

The modified step 3′ restricts the message dissemination inside the embedded $\mathrm{K}(d, l)$. Hence, the following lemma holds as a consequence of Lemma 3.1.

LEMMA 3.2. *Starting from any node of an embedded* $\mathrm{K}(d, l)$, *where* $1 \leqslant l < t$, *the modified Chen and Duh's algorithm can disseminate a message to each node of the embedded* $\mathrm{K}(d, l)$ *exactly once. Moreover, the resulting spanning tree has height at most* $2^l - 1$.

In the following we show that all spanning trees for $\mathrm{K}(d, t)$ induced by Chen and Duh's algorithm have minimum heights.

LEMMA 3.3. *When applying Chen and Duh's algorithm to* $\mathrm{K}(d, t)$, *all t-frontiers can receive the message from the source node r via the shortest paths.*

Proof. It is trivial to see that this lemma holds for $t = 1, 2$. Hence we assume the lemma holds for $t \leqslant s$, where $s \geqslant 2$ is a positive integer. We now consider the situation of $t = s+1$. Without loss of generality, assume $r \in x \cdot \mathrm{K}(d, s)$, where $0 \leqslant x \leqslant d-1$. By assumption the $(s+1)$-frontier $(x)^{s+1}$ can receive the message from r via the shortest path. According to Chen and Duh's broadcasting algorithm, the message enters each $y \cdot \mathrm{K}(d, s)$, where $0 \leqslant y \leqslant d-1$ and $y \neq x$, via the s-flipping link $(x(y)^s, y(x)^s)$. So, the transmission path from r to each $(s+1)$-frontier $(y)^{s+1}$, where $0 \leqslant y \leqslant d-1$ and $y \neq x$, can be expressed as $r \to \cdots \to x(y)^s \to y(x)^s \to \cdots \to (y)^{s+1}$, where the two subpaths $r \to \cdots \to x(y)^s$ and $y(x)^s \to \cdots \to (y)^{s+1}$ are the shortest by assumption. The entire transmision path can be guaranteed the shortest if the shortest path from r to $(y)^{s+1}$ contains a unique s-flipping link, i.e., $(x(y)^s, y(x)^s)$. The latter can be easily proved by contradiction as follows.

Suppose the shortest path from r to $(y)^{s+1}$ passes through $z \cdot \mathrm{K}(d, s)$, where $0 \leqslant z \leqslant d-1$, $z \neq x$, and $z \neq y$. Since any two s-frontiers within the same embedded $\mathrm{K}(d, s)$ have a distance of $2^s - 1$ (see [4]), the length of the shortest path is at least $1 + (2^s - 1) + 1 + (2^s - 1) = 2^{s+1}$, which contradicts the diameter of $\mathrm{K}(d, s+1)$. The latter was proved to be $2^{s+1} - 1$ in [4]. Q.E.D.

LEMMA 3.4. *All spanning trees for* $\mathrm{K}(d, t)$ *induced by Chen and Duh's algorithm have minimum heights.*

Proof. Without loss of generality, assume the source node $r \in x \cdot \mathrm{K}(d, t-1)$, where $0 \leqslant x \leqslant d-1$, and T is the spanning tree induced by Chen and Duh's broadcasting algorithm. Let F be the set of the farthest nodes from r in T. Lemma 3.2 assures us that all nodes in F are outside $x \cdot \mathrm{K}(d, t-1)$. Suppose conversely that T does not have minimum height. Then, for each node $p \in F$ we have $l(r, p) > d(r, p)$, where $l(r, p)$ is the length of the path from r to p in T. Without loss of generality, assume $p \in y \cdot \mathrm{K}(d, t-1)$, where $0 \leqslant y \leqslant d-1$ and $y \neq x$. According to Chen and Duh's broadcasting algorithm, we have $l(r, p) \leqslant l(r, (y)^t)$, which means that the t-frontier $(y)^t$ also belongs to F. However, by Lemma 3.3 we have $l(r, (y)^t) = d(r, (y)^t)$. This is a contradiction. Q.E.D.

Now we are going to construct $ST(m, n, r, m)$ in $\mathrm{IK}(d, t)$ $(= S_{t-1} + S_{t-2} + \cdots + S_i)$, where $t-1 \geqslant m \geqslant n \geqslant i$. Recall that S_m contains b_m embedded $\mathrm{K}(d, m)$'s that are completely connected by m-flipping links. Moreover, since $r \in S_m$, we have $r = r_{t-1} r_{t-2} \cdots r_{m+1} r_m \cdots r_1 r_0 \in b_{t-1} b_{t-2} \cdots b_{m+1} r_m \cdots r_1 r_0 \in b_{t-1} b_{t-2} \cdots b_{m+1} r_m \cdot \mathrm{K}(d, m)$, where $0 \leqslant r_m \leqslant b_m - 1$. By the aid of the modified Chen and Duh's algorithm, $ST(m, m, r, m)$, i.e., a spanning tree of S_m rooted at node r, can be constructed as the union of the components:

- A spanning tree of $b_{t-1} b_{t-2} \cdots b_{m+1} r_m \cdot \mathrm{K}(d, m)$ rooted at node r.
- Link set $\{(b_{t-1} b_{t-2} \cdots b_{m+1} r_m (x)^m,\ b_{t-1} b_{t-2} \cdots b_{m+1} x (r_m)^m) \mid 0 \leqslant x \leqslant b_m - 1 \text{ and } x \neq r_m\}$.
- Spanning trees of $b_{t-1} b_{t-2} \cdots b_{m+1} x \cdot \mathrm{K}(d, m)$ rooted at $b_{t-1} b_{t-2} \cdots b_{m+1} x (r_m)^m$ for all $0 \leqslant x \leqslant b_m - 1$ and $x \neq r_m$.

Constructing $ST(m, n, r, m)$ for $m > n$ proceeds with examining $S_m + S_{m-1} + \cdots + S_n$ from the left to the right and recursively executing the following five dissemination patterns.

Pattern A. If $r_m \leqslant b_{m-1} - 1$, then $ST(m, n, r, m)$ is constructed as the union of the components (refer to Fig. 4(a)).

- $ST(m, m, r, m)$.
- Link $(b_{t-1} b_{t-2} \cdots b_{m+1} r_m (b_m)^m,\ b_{t-1} b_{t-2} \cdots b_{m+1} b_m (r_m)^m)$.
- $ST(m-1, n, r', m-1)$, where $r' = b_{t-1} b_{t-2} \cdots b_{m+1} b_m (r_m)^m$.

Pattern B. If $r_m = b_{m-1}$ and there exists one jumping flipping link from S_m to some S_l (i.e., $J^{b_{m-1}}_{m,l}$), where $m-1 > l \geqslant n$, then by Theorem 2.1 we have $b_m > b_{m-1} = b_{m-2} = \cdots = b_{l+1} < b_l$, and $ST(m, n, r, m)$ is constructed as the union of the components (refer to Fig. 4(b)):

- $ST(m, m, r, m)$.
- Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{m+1}r_m(b_m)^m, b_{t-1}b_{t-2}\cdots b_{m+1}b_m(r_m)^m)$.
- $ST(m-1, n, r', l)$, where $r' = b_{t-1}b_{t-2}\cdots b_{m+1}b_m(r_m)^m = b_{t-1}b_{t-2}\cdots b_{l+1}(r_m)^{l+1}$.

Pattern C. If $r_m > b_{m-1}$ and there exists one jumping flipping link from S_m to some S_l, where $m-1 > l \geqslant n$, then $ST(m, n, r, m)$ is constructed as the union of the components (refer to Fig. 4(c)):

- $ST(m, m, r, m)$.
- Link set $\{(b_{t-1}b_{t-2}\cdots b_{m+1}x(b_m)^m, b_{t-1}b_{t-2}\cdots b_{m+1}b_m(x)^m) \mid 0 \leqslant x \leqslant b_{m-1}-1\}$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{m+1}b_m x \cdot \mathrm{K}(d, m-1)$ rooted at $b_{t-1}b_{t-2}\cdots b_{m+1}b_m(x)^m$ for all $0 \leqslant x \leqslant b_{m-1}-1\}$.
- Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{m+1}b_{m-1}(b_m)^m, b_{t-1}b_{t-2}\cdots b_{m+1}b_m(b_{m-1})^m)$.
- $ST(m-2, n, r', l)$, where $r' = b_{t-1}b_{t-2}\cdots b_{m+1}b_m(b_{m-1})^m = b_{t-1}b_{t-2}\cdots b_{l+1}(b_{m-1})^{l+1}$.

Pattern D. If $r_m \geqslant b_{m-1}$ and there is no jumping flipping link from S_m to some S_l, where $m-1 > l \geqslant n$, then we determine the leftmost jumping flipping link, say $J^e_{u,v}$, in $S_m + S_{m-1} + \cdots + S_n$, if it exists. Since $J^e_{u,v}$ is the leftmost one, by Theorem 2.1 we have $b_m > (r_m \geqslant) b_{m-1} \geqslant b_{m-2} \geqslant \cdots \geqslant b_u > b_{u-1} = b_{u-2} = \cdots = b_{v+1} < b_v$ and $e = b_{u-1} = b_{u-2} = \cdots = b_{v+1}$. Then $ST(m, n, r, m)$ is constructed as the union of the components (refer to Fig. 4(d)):

- $ST(m, m, r, m)$.
- Link sets $\{(b_{t-1}b_{t-2}\cdots b_{j+1}x(b_j)^j, b_{t-1}b_{t-2}\cdots b_{j+1}b_j(x)^j) \mid 0 \leqslant x \leqslant b_{j-1}-1\}$ for all $u \leqslant j \leqslant m$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots \cdot b_{j+1}b_j x \cdot \mathrm{K}(d, j-1)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+1}b_j(x)^j$ for all $0 \leqslant x \leqslant b_{j-1}-1$ and all $u \leqslant j \leqslant m$.
- Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{u+1}e(b_u)^u, b_{t-1}b_{t-2}\cdots b_{u+1}b_u(e)^u)$.
- $ST(u-2, n, r', v)$, where $r' = b_{t-1}b_{t-2}\cdots b_{u+1}b_u(e)^u = b_{t-1}b_{t-2}\cdots b_{v+1}(e)^{v+1}$.

If $J^e_{u,v}$ does not exist, then $ST(m, n, r, m)$ is constructed as the union of the first three components above, with substituting $n+1$ for u.

Note that the last components in *Patterns* B, C, and D are spanning trees of the form $ST(m', n, r', k)$, where $m' = k$ or $m' > k$ and $b_{m'} = b_{m'-1} = \cdots = b_{k+1} < b_k$. If

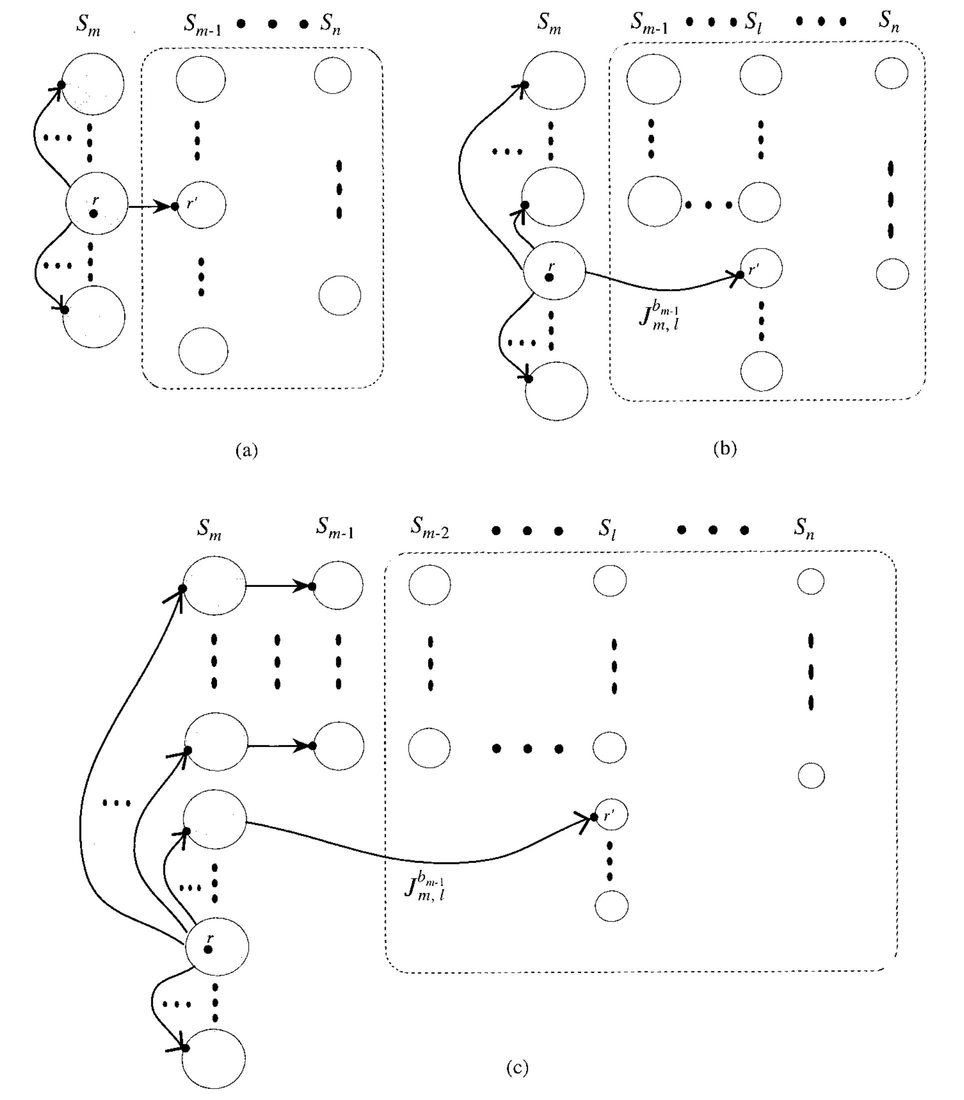
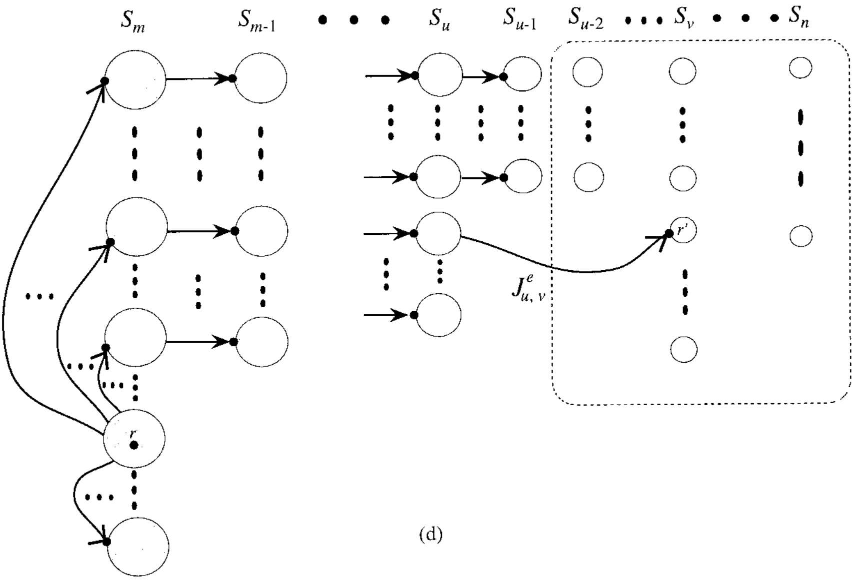


FIG. 4. Five dissemination patterns for constructing $ST(m, n, r, m)$: (a) *Pattern* A, (b) *Pattern* B, (c) *Pattern* C, (d) *Pattern* D, (e) *Pattern* E.

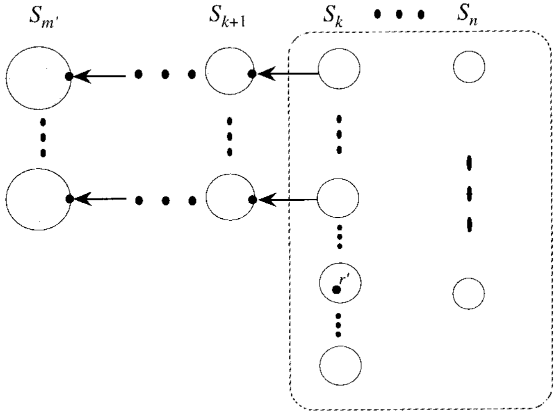
$m' = k$, $ST(m', n, r', k)$ is constructed recursively. Otherwise, $ST(m', n, r', k)$ is constructed as

Pattern E. $ST(m', n, r', k)$, where $b_{m'} = b_{m'-1} = \cdots = b_{k+1} < b_k$ and $m' > k \geqslant n$, is constructed as the union of the components (assuming $h = b_{m'} = b_{m'-1} = \cdots = b_{k+1}$) (refer to Fig. 4(e)):

- $ST(k, n, r', k)$.
- Link sets $\{(b_{t-1}b_{t-2}\cdots b_{j+1}b_j(x)^j,\ b_{t-1}b_{t-2}\cdots b_{j+1}x(b_j)^j \mid 0 \leqslant x \leqslant h-1\}$ for all $k < j \leqslant m'$.



(d)



(e)

FIG. 4—*Continued*

- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{j+1}x \cdot \mathrm{K}(d, j)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+1}x(b_j)^j$ for all $0 \leqslant x \leqslant h-1$ and all $k < j \leqslant m'$.

With these patterns, $ST(m, n, r, m)$, where $m > n$, can be constructed recursively. For example, let us consider $\mathrm{IK}(d, 11)$ with $d > 5$ and coefficient vector (4, 5, 4, 2, 2, 4, 2, 2, 1, 5, $*$). Assuming $r \in 453 \cdot \mathrm{K}(d, 8)$, $ST(8, 2, r, 8)$ can be constructed as follows (refer to Fig. 5(a)). First, *Pattern* C is applied because $r_8 = 3 > 2 = b_7$ and there exists one jumping flipping link from S_8 to S_5. A spanning tree of $S_8 + S_7$ rooted at r is thus obtained. Then *Pattern* E is applied to construct $ST(6, 2, r', 5)$, where $r' = 454(2)^8 = 45422(2)^6$, and the spanning tree of $S_8 + S_7$ rooted at r grows by augmenting $ST(5, 2, r', 5)$ and the nodes of S_6. $ST(5, 2, r', 5)$ can be obtained by applying *Pattern* D.

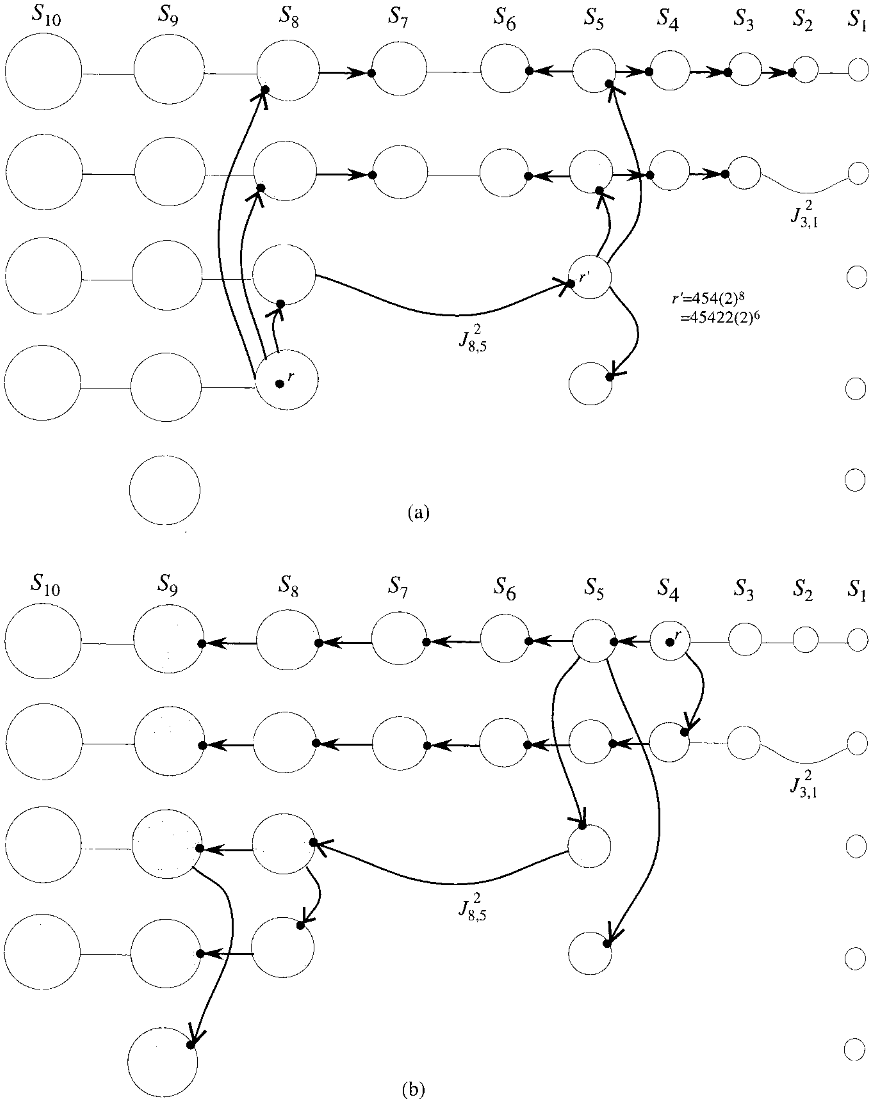


FIG. 5. Examples: (a) $ST(8, 2, r, 8)$, where r belongs to $453 \cdot \mathrm{K}(d, 8)$; (b) $ST(9, 4, r, 4)$, where r belongs to $4542240 \cdot \mathrm{K}(d, 4)$.

4. CONSTRUCTING $ST(m, n, r, n)$

In this section, we aim to construct $ST(m, n, r, n)$. The construction algorithm contains three basic dissemination patterns: *Pattern* F, *Pattern* G, and *Pattern* H. A variable w is used by the algorithm. Initially, w is set to r_n, where $r = r_{t-1}r_{t-2}\cdots r_1r_0$ is assumed, and $ST(n, n, r, n)$ is constructed as an initial spanning tree. The current spanning tree, assuming $ST(j, n, r, n)$ and $n \leqslant j < m$, will grow toward the left when the patterns are applied. The details are shown.

Pattern F. If $b_j \leqslant b_{j+1}$, then $ST(j, n, r, n)$ grows into $ST(j+1, n, r, n)$ by augmenting the components (refer to Fig. 6(a)):

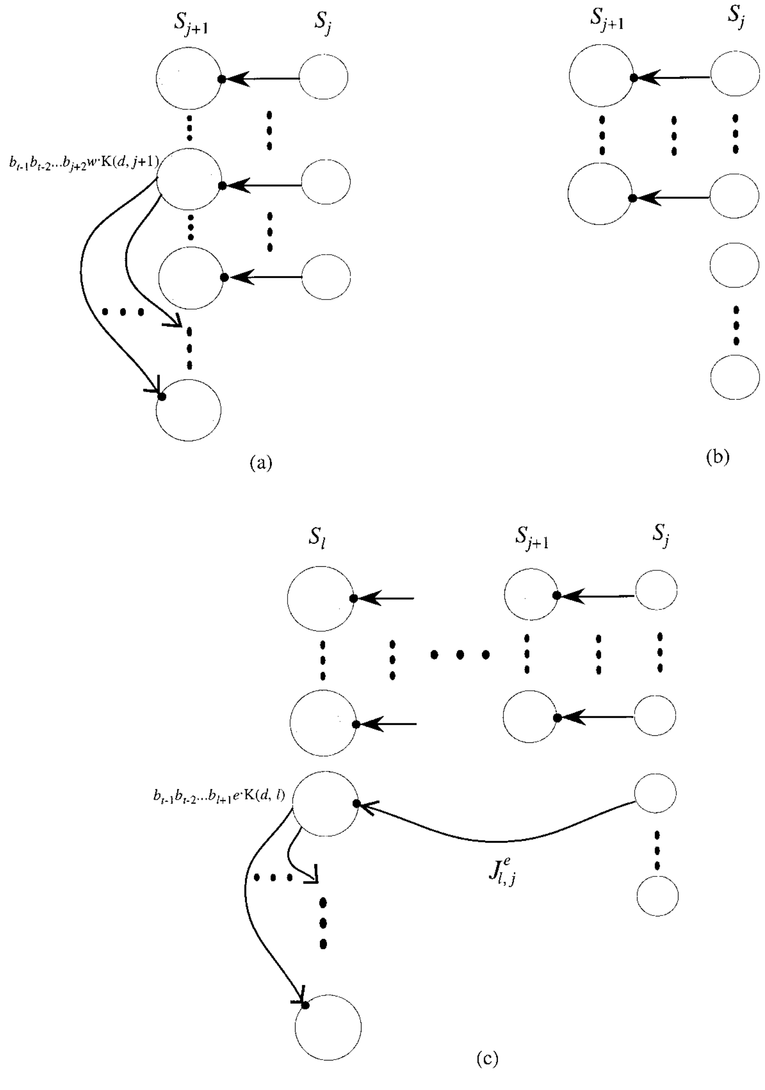


FIG. 6. Three dissemination patterns for constructing $ST(m, n, r, n)$: (a) *Pattern* F, (b) *Pattern* G, (c) *Pattern* H.

• Link set $\{(b_{t-1}b_{t-2}\cdots b_{j+2}b_{j+1}(x)^{j+1},\ b_{t-1}b_{t-2}\cdots b_{j+2}x(b_{j+1})^{j+1}) \mid 0 \leqslant x \leqslant b_j - 1\}$.

• Spanning trees of $b_{t-1}b_{t-2}\cdots b_{j+2}x \cdot \mathrm{K}(d, j+1)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+2}x(b_{j+1})^{j+1}$ for all $0 \leqslant x \leqslant b_j - 1$.

• Link set $\{(b_{t-1}b_{t-2}\cdots b_{j+2}w(y)^{j+1},\ b_{t-1}b_{t-2}\cdots b_{j+2}y(w)^{j+1}) \mid b_j \leqslant y \leqslant b_{j+1} - 1\}$.

• Spanning trees of $b_{t-1}b_{t-2}\cdots b_{j+2}y \cdot \mathrm{K}(d, j+1)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+2}y(w)^{j+1}$ for all $b_j \leqslant y \leqslant b_{j+1} - 1$.

The subnetwork $b_{t-1}b_{t-2}\cdots b_{j+2}w \cdot \mathrm{K}(d, j+1)$ is responsible for disseminating the message to $b_{t-1}b_{t-2}\cdots b_{j+2}y \cdot \mathrm{K}(d, j+1)$ for all $b_j \leqslant y \leqslant b_{j+1} - 1$. They are not allowed to receive the message directly from S_j.

Pattern G. If $b_j > b_{j+1}$ and there is no jumping flipping link from S_j to some S_l, where $m \geqslant l > j+1$, then $ST(j, n, r, n)$ grows into $ST(j+1, n, r, n)$ by augmenting the components (refer to Fig. 6(b)):

• Link set $\{(b_{t-1}b_{t-2}\cdots b_{j+2}b_{j+1}(x)^{j+1},\ b_{t-1}b_{t-2}\cdots b_{j+2}x(b_{j+1})^{j+1}) \mid 0 \leqslant x \leqslant b_{j+1}-1\}$.

• Spanning trees of $b_{t-1}b_{t-2}\cdots b_{j+2}x \cdot \mathrm{K}(d, j+1)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+2}x(b_{j+1})^{j+1}$ for all $0 \leqslant x \leqslant b_{j+1}-1$.

Pattern H. If $b_j > b_{j+1}$ and there exists one jumping flipping link from S_j to some S_l, where $m \geqslant l > j+1$, then $ST(j, n, r, n)$ grows into $ST(l, n, r, n)$ by augmenting the components (refer to Fig. 6(c), where $e = b_{l-1} = b_{l-2} = \cdots = b_{j+1}$ is assumed:

• Link sets $\{(b_{t-1}b_{t-2}\cdots b_{s+1}b_s(x)^s,\ b_{t-1}b_{t-2}\cdots b_{s+1}x(b_s)^s) \mid 0 \leqslant x \leqslant e-1\}$ for all $j < s \leqslant l$.

• Spanning trees of $b_{t-1}b_{t-2}\cdots b_{s+1}x \cdot \mathrm{K}(d, s)$ rooted at $b_{t-1}b_{t-2}\cdots b_{s+1}x(b_s)^s$ for all $0 \leqslant x \leqslant e-1$ and all $j < s \leqslant l$.

• Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{l+1}b_l(e)^l,\ b_{t-1}b_{t-2}\cdots b_{l+1}e(b_l)^l)$.

• Spanning tree of $b_{t-1}b_{t-2}\cdots b_{l+1}e \cdot \mathrm{K}(d, l)$ rooted at $b_{t-1}b_{t-2}\cdots b_{l+1}e(b_l)^l$.

• Link set $\{(b_{t-1}b_{t-2}\cdots b_{l+1}e(y)^l,\ b_{t-1}b_{t-2}\cdots b_{l+1}y(e)^l) \mid e+1 \leqslant y \leqslant b_l-1\}$.

• Spanning trees of $b_{t-1}b_{t-2}\cdots b_{l+1}y \cdot \mathrm{K}(d, l)$ rooted at $b_{t-1}b_{t-2}\cdots b_{l+1}y(e)^l$ for all $e+1 \leqslant y \leqslant b_l-1$.

Besides, w is updated to e after *Pattern* H is executed. The purpose of this change is to maintain the height of $ST(m, n, r, n)$ as small as possible.

By the aid of the three patterns, $ST(m, n, r, n)$ can be constructed from the right to the left. For example, refer to Fig. 5(b), where $r \in 4542240 \cdot \mathrm{K}(d, 4)$ and $ST(9, 4, r, 4)$ is shown. The $\mathrm{IK}(d, t)$ has $d > 5$, $t = 11$, and coefficient vector (4, 5, 4, 2, 2, 4, 2, 2, 1, 5, $*$). Initially, $ST(4, 4, r, 4)$ is constructed as an initial spanning tree and w is set to $r_4 = 0$. Since $b_4 = 2 < 4 = b_5$, *Pattern* F is first applied and $ST(4, 4, r, 4)$ grows into $ST(5, 4, r, 4)$. Next, *Pattern* H is applied and the current spanning tree becomes $ST(8, 4, r, 4)$. Also w is updated to 2 ($= b_6 = b_7$). Finally, $ST(9, 4, r, 4)$ results after *Pattern* F is applied.

5. CONSTRUCTING $ST(m, n, r, z)$

In this section, we explain how to broadcast in $\mathrm{IK}(d, t)$ by constructing $ST(m, n, r, z)$, where $m \leqslant z \leqslant n$. Since the two cases of $z = m$ and $z = n$ have been discussed in the previous two sections, we only need to consider $m > z > n$. Without loss of generality, we assume $r \in b_{t-1}b_{t-2}\cdots b_{z+1}\alpha \cdot \mathrm{K}(d, z)$, where $0 \leqslant \alpha \leqslant b_z - 1$. First of all, we have to determine whether or not there is a jumping flipping link passing S_z, i.e., whether or not some $J^e_{u,v}$ exists such that $m \geqslant u > z > v \geqslant n$. If not, $ST(m, n, r, z)$

can be obtained by first constructing $ST(z, n, r, z)$ and then expanding it to $ST(m, n, r, z)$. The former can be done in the same way as described in Section 3. The latter can be done just like in Section 4, but $ST(z, n, r, z)$, instead of $ST(z, z, r, z)$, is regarded as the initial spanning tree, and w is initialized with α.

If such a $J^e_{u,v}$ exists, by Theorem 2.1 we have $b_u > b_{u-1} = b_{u-2} = \cdots = b_{v+1} < b_v$ and $e = b_{u-1} = b_{u-2} = \cdots = b_{v+1}$. $ST(m, n, r, z)$ is constructed according to two cases:

Case 1 ($z = u-1$). First, $ST(u, n, r, z)$ is obtained by combining together the components (refer to Fig. 7(a), where $\alpha = e-1$ is assumed):

- $ST(z, n, r, z)$.
- Link set $\{(b_{t-1}b_{t-2}\cdots b_{u+1}b_u(x)^u, b_{t-1}b_{t-2}\cdots b_{u+1}x(b_u)^u) \mid 0 \leqslant x \leqslant e-1\}$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{u+1}x \cdot \mathrm{K}(d, u)$ rooted at $b_{t-1}b_{t-2}\cdots b_{u+1}x(b_u)^u$ for all $0 \leqslant x \leqslant e-1$.
- Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{u+1}b_u(e)^u, b_{t-1}b_{t-2}\cdots b_{u+1}e(b_u)^u)$.
- Spanning tree of $b_{t-1}b_{t-2}\cdots b_{u+1}e \cdot \mathrm{K}(d, u)$ rooted at $b_{t-1}b_{t-2}\cdots b_{u+1}e(b_u)^u$.
- Link set $\{(b_{t-1}b_{t-2}\cdots b_{u+1}\alpha(y)^u, b_{t-1}b_{t-2}\cdots b_{u+1}y(\alpha)^u) \mid e+1 \leqslant y \leqslant b_u - 1\}$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{u+1}y \cdot \mathrm{K}(d, u)$ rooted at $b_{t-1}b_{t-2}\cdots b_{u+1}y(\alpha)^u$ for all $e+1 \leqslant y \leqslant b_u - 1$.

The subnetwork $b_{t-1}b_{t-2}\cdots b_{u+1}e \cdot \mathrm{K}(d, u)$ in S_u receives the message via $J^e_{u,v}$ in order to reduce the transmission length. Then, $ST(u, n, r, z)$ grows into $ST(m, n, r, z)$ by augmenting the nodes of $S_m + S_{m-1} + \cdots + S_{u+1}$, which can be done almost the same as in Section 4, except that $ST(u, n, r, z)$ is regarded as the initial spanning tree and w is initialized with α.

Case 2 ($z < u-1$). First, $ST(u, n, r, z)$ is obtained by combining together the components (refer to Fig. 7(b), where $\alpha = e-1$ is assumed):

- $ST(z, n, r, z)$.
- Link sets $\{(b_{t-1}b_{t-2}\cdots b_{j+1}b_j(x)^j, b_{t-1}b_{t-2}\cdots b_{j+1}x(b_j)^j) \mid 0 \leqslant x \leqslant e-1\}$ for all $z < j \leqslant u$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{j+1}x \cdot \mathrm{K}(d, j)$ rooted at $b_{t-1}b_{t-2}\cdots b_{j+1}x(b_j)^j$ for all $0 \leqslant x \leqslant e-1$ and all $z < j \leqslant u$.
- Jumping flipping link $(b_{t-1}b_{t-2}\cdots b_{u+1}b_u(e)^u, b_{t-1}b_{t-2}\cdots b_{u+1}e(b_u)^u)$.
- Spanning tree of $b_{t-1}b_{t-2}\cdots b_{u+1}e \cdot \mathrm{K}(d, u)$ rooted at $b_{t-1}b_{t-2}\cdots b_{u+1}e(b_u)^u$.
- Link set $\{(b_{t-1}b_{t-2}\cdots b_{u+1}e(y)^u, b_{t-1}b_{t-2}\cdots b_{u+1}y(e)^u) \mid e+1 \leqslant y \leqslant b_u - 1\}$.
- Spanning trees of $b_{t-1}b_{t-2}\cdots b_{u+1}y \cdot \mathrm{K}(d, u)$ rooted at $b_{t-1}b_{t-2}\cdots b_{u+1}y(e)^u$ for all $e+1 \leqslant y \leqslant b_u - 1$.

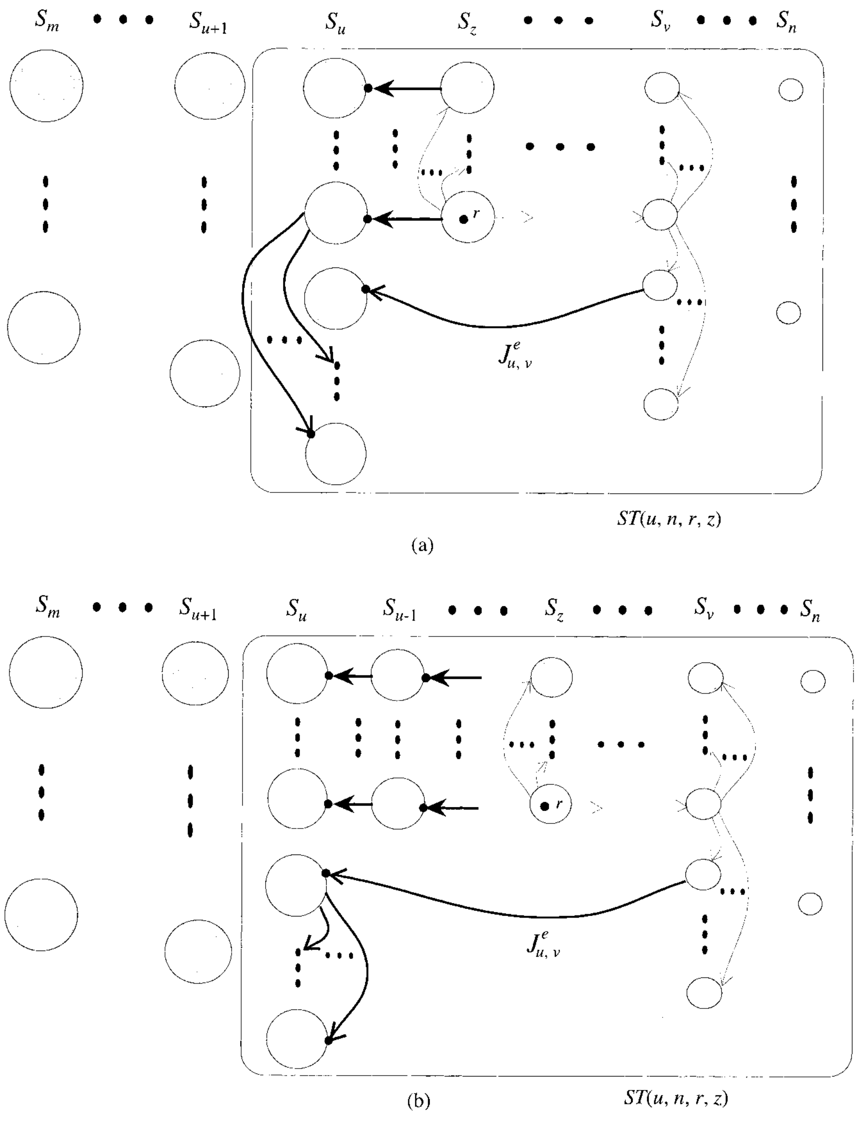


FIG. 7. Illustration of $ST(m, n, r, z)$: (a) $z = u - 1$, (b) $z < u - 1$.

Like Case 1, $ST(u, n, r, z)$ then grows into $ST(m, n, r, z)$ by augmenting the nodes of $S_m + S_{m-1} + \cdots + S_{u+1}$ with $ST(u, n, r, z)$ being the initial spanning tree and e being the initial value of w.

A distributed algorithm (in pseudo codes) for broadcasting on IK(d, t) can be found in [32].

6 EXPERIMENTS AND RESULTS

Clearly our broadcasting algorithm achieves optimal message complexity, because each node receives the message exactly once. Besides, extensive experiments had been made to verify the efficiency. Four algorithms, as well as our broadcasting algorithm, were implemented for the need of our experiments:

• Su, Chen, and Duh's algorithm for computing the diameter of $\mathrm{IK}(d, t)$ [31]. Using the prune-and-search technique [20], the algorithm can compute the diameter of $\mathrm{IK}(d, t)$ in $O(t)$ time. Moreover, the farthest pair of nodes can be determined simultaneously. Note that, although the diameter of $\mathrm{IK}(d, t)$ can be computed, no formula is available for computing it.

• Su, Chen, and Duh's shortest-path routing algorithm for $\mathrm{IK}(d, t)$ [30]. With $O(d \cdot t)$ time preprocessing performed by the source node, the algorithm takes $O(t)$ time for each intermediate node to determine the next node along the shortest path to the destination node.

• Chen and Duh's shortest-path routing algorithm for $\mathrm{K}(d, t)$ [4]. The algorithm can transmit a message from p to q in $O(t + d(p, q))$ time, where p and q are any two nodes of $\mathrm{K}(d, t)$ and $d(p, q)$ is their distance.

• Chen and Duh's broadcasting algorithm for $\mathrm{K}(d, t)$ [4]. The algorithm was reviewed in the beginning of Section 3.

All the algorithms above were simulated using C programs on the PC. We first compared the maximum transmission length, i.e., the height of $ST(t-1, i, r, z)$, with the diameter of $\mathrm{IK}(d, t)$. Table 1 shows the experimental result. For each of the entries marked with $*$, the experiment was made exhaustively. That is, for each $d^{t-1} < N < d^t$, we let each of the N nodes act as the source node and then determine the height of the corresponding $ST(t-1, i, r, z)$. Our experimental result shows that all $ST(t-1, i, r, z)$'s have their heights bounded above by the diameter. The diameter was computed by Su *et al.*'s algorithm [31].

On the other hand, for each of the entries marked with Δ, we randomly chose 10^5 experiment instances (instances for short). Here, an arbitrarily selected value of N combined with an arbitrarily selected source node r forms an instance. Our experimental result shows that their corresponding $ST(t-1, i, r, z)$'s have their heights bounded above by the diameters. Moreover, the experiment was also made by selecting peripheral nodes as r. A node in a network is said to be a *peripheral node* [3] if the distance between it and its farthest node is equal to the diameter of

TABLE 1

Experimental Results about Maximum Transmission Length

d \ t	2	3	4	5	6	7	8	9	10	11	12
3	*	*	*	*	*	*	*	*	*	△	△
4	*	*	*	*	*	*	*	*	△	△	△
5	*	*	*	*	*	*	△	△	△	△	△
6	*	*	*	*	*	△	△	△	△	△	△
7	*	*	*	*	*	△	△	△	△	△	△
8	*	*	*	*	△	△	△	△	△	△	△
9	*	*	*	*	△	△	△	△	△	△	△
10	*	*	*	△	△	△	△	△	△	△	△

the network. Su *et al.*'s algorithm [31] can find two peripheral nodes as well. It is clear that the height of $ST(t-1, i, r, z)$ is at least the diameter if r is a peripheral node. Our experimental result shows that the height of $ST(t-1, i, r, z)$ for peripheral node r is just the same as the diameter.

In addition to the maximum transmission length, we also investigated how close to the shortest paths the transmission paths are. Figure 8 shows the average percentages of nodes that receive the message from the source node via the shortest paths. Chen and Duh's broadcasting algorithm [4] for $\mathrm{K}(d, t)$ was adopted in the experiment. The experiment proceeded as follows. First, 1000 instances were randomly chosen for both $\mathrm{IK}(d, t)$ and $\mathrm{K}(d, t)$. Here a selected value of N combined with a selected source node r forms an instance of $\mathrm{IK}(d, t)$, whereas a selected source node alone forms an instance of $\mathrm{K}(d, t)$. For each of the chosen instances, the percentage was computed by the aid of Su *et al.*'s algorithm [30] and Chen and Duh's algorithm [4]. The two algorithms can compute the distance of two arbitrary nodes in $\mathrm{IK}(d, t)$ and $\mathrm{K}(d, t)$, respectively. The average percentages for $\mathrm{IK}(d, t)$ and $\mathrm{K}(d, t)$ were then computed for the 1000 instances (we found in the experiment that all average percentages got stable after running 1000 instances).

For illustration, we show the experimental results for an instance of IK(6, 8) and an instance of K(6, 8). We let 30120441 be the source node, and $N = 1506648$. The simulation program output 1,432,230 nodes and 1,578,344 nodes for the instance of IK(6, 8) and K(6, 8), respectively, which received the messages via the shortest paths. The percentages are 0.9506 and 0.9397, respectively.

The interested readers may wonder why the curves for $\mathrm{IK}(d, t)$ are above the curves for $\mathrm{K}(d, t)$ in Fig. 8. Here we try to explain it with an example as follows. Let us consider K(4, 3) and those IK(4, 3)'s with coefficient vectors $(2, *)$, $(2, 1, *)$, $(2, 2, *)$, $(2, 3, *)$, $(3, *)$, and $(3, 1, *)$, respectively. The shortest path between nodes 032 and 133 in K(4, 3) is $032 \rightarrow 033 \rightarrow 300 \rightarrow 301 \rightarrow 310 \rightarrow 311 \rightarrow 133$, which has length 6. However, this shortest path is not existent again for those IK(4, 3)'s. Instead, the distance between 032 and 133 increases to 7 for those IK(4, 3)'s.

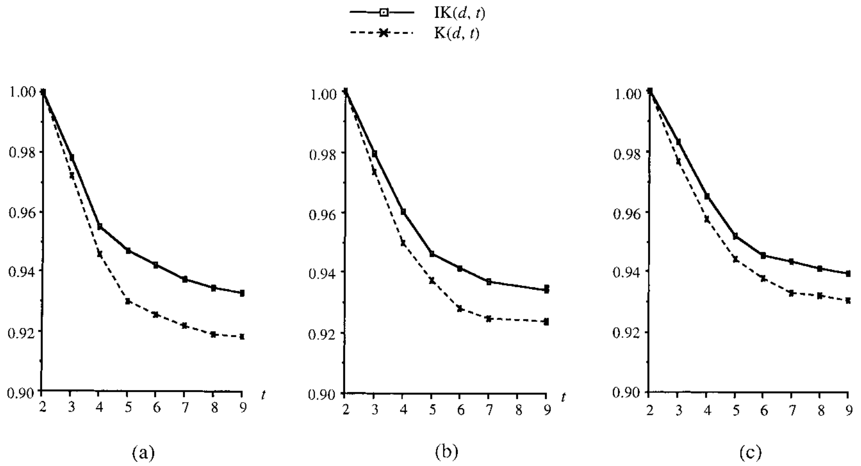


FIG. 8. Average percentages of nodes that receive the message via the shortest paths: (a) $d = 4$: (b) $d = 5$; (c) $d = 6$.

Assume 032 is the source node. The transmission path from 032 to 133 in K(4, 3) induced by Chen and Duh's broadcasting algorithm [4] has length 7. On the other hand, the transmision paths from 032 to 133 in those IK(4, 3)'s induced by our broadcasting algorithm all have length 7, which is the distance between 032 and 133 in those IK(4, 3)'s.

Also a careful reader may find that the curves in Fig. 8 go upward as d increases. This is a consequence of the following observation. Assume r belongs to some K(d, z) of S_z. The closer to some z-frontier the root node r is, the smaller the average percentage is. For example, refer to Fig. 2 again, where the structure of IK(4, 3) with coefficient vector $(3, 2, *)$ is shown. If $r = 032$, which is one distant from the nearest 2-frontier (i.e., 033), then all nodes but 133 in the IK(4, 3) can receive the message from r via the shortest paths. On the other hand, if $r = 033$ is a 2-frontier, then three nodes 133, 132, and 131 cannot receive the message from r via the shortest paths. Since the expected distance between r and the nearest z-frontier increases as d grows, the curves in Fig. 8 go upward as d grows.

Figure 9 further shows the average deviations for the nonshortest transmission paths. Similar to Fig. 8, 1000 randomly chosen instances were executed for both IK(d, t) and K(d, t). For each nonshortest transmission path (from r to v, for example), the deviation was computed as the ratio of its length to $d(r, v)$. The average deviations were then obtained each by taking the average of all deviations of nonshortest transmission paths. Also note that the curves start with $t = 3$ because the transmission paths for $t = 2$ are all shortest.

In Table 1 we have investigated the height of $ST(t-1, i, r, z)$. However, since different root nodes will lead to different tree heights, we are more concerned with the percentage of $ST(t-1, i, r, z)$'s that have minimum heights. Figure 10 shows our experimental result. Each percentage value for IK(d, t) was obtained by running 12,000 randomly chosen instances. It is observed that a high percentage of $ST(t-1, i, r, z)$'s have minimum heights. Also note that all spanning trees for K(d, t) induced by Chen and Duh's broadcasting algorithm [4] have minimum heights. This can be assured by Lemma 3.4.

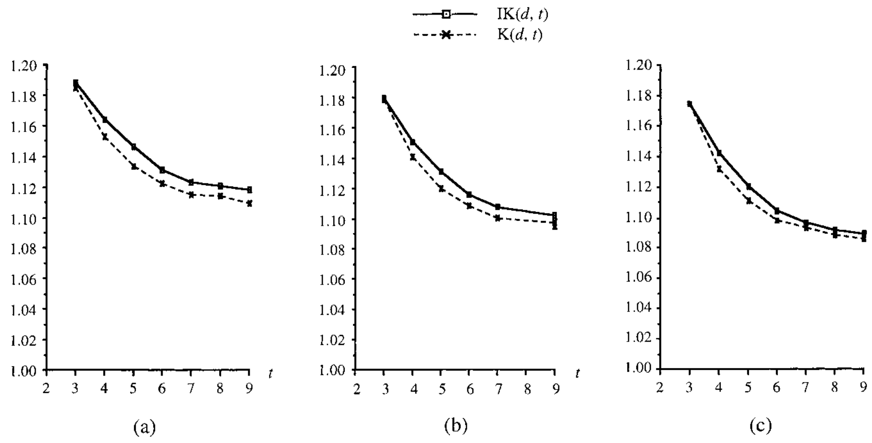


FIG. 9. Average deviations for nonshortest transmission paths: (a) $d = 4$; (b) $d = 5$; (c) $d = 6$.

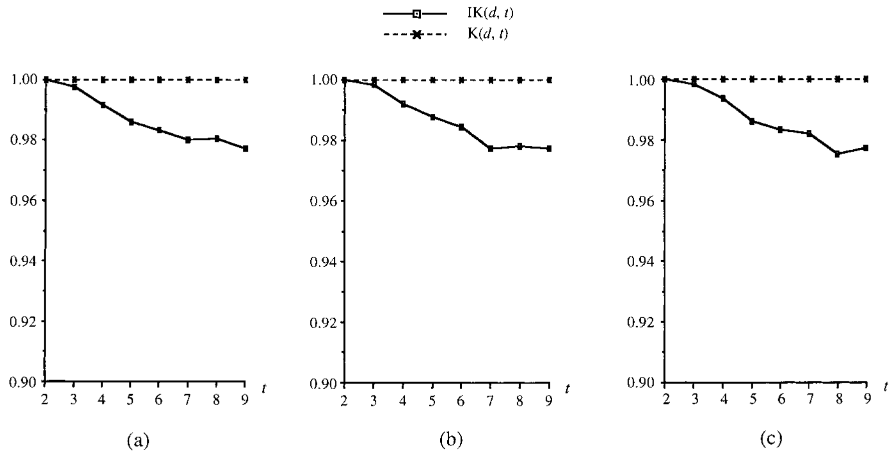


FIG. 10. Percentages of $ST(t-1, i, r, z)$'s with minimum heights: (a) $d=4$; (b) $d=5$; (c) $d=6$.

7 CONCLUDING REMARKS

Almost all announced (complete) networks have suffered from a common problem: the number of nodes is restricted to a set of specific values. Hence, several incomplete networks, such as incomplete hypercubes [17], incomplete star networks [18, 25], incomplete rotator graphs [23], and incomplete WK-recursive networks [30], have been suggested as a solution to the problem.

Generally speaking, broadcasting on incomplete networks is more difficult than on corresponding complete networks. For example, although several broadcasting algorithms [22, 28] for the star networks have been proposed, the existing one [18] for the incomplete star networks was designed only for those with size $c \cdot k!$, where $1 < c \leqslant k$. Furthermore, no broadcasting algorithms for the incomplete rotator graphs and the incomplete WK-recursive networks were designed before. We think, based on our experience of studying incomplete networks, the difficulty has arisen mainly from the lack of unified representations for incomplete networks. Unlike complete networks, incomplete networks of different sizes have a significant difference in their topologies. For example, $\mathrm{K}(d, t)$ looks very similar to $\mathrm{K}(d, t-1)$, whereas two $\mathrm{IK}(d, t)$'s with different sizes may look very unlike in their topologies. Thus, a unified representation is very helpful to the study of the incomplete WK-recursive networks.

In this paper we have shown that the incomplete WK-recursive networks can be conveniently represented with the multistage graphs, and thus broadcasting on the incomplete WK-recursive networks is equivalent to constructing spanning trees for the corresponding multistage graphs. The resulting broadcasting algorithm achieves optimal message complexity. Besides, experimental results showed that (1) the maximum transmission length does not exceed the diameter, (2) a high percentage of nodes can receive the message from the source node via the shortest paths, (3) the deviations for those nonshortest transmission paths are small, and (4) a high percentage of the broadcasting trees have minimum heights.

One of our further researches is first to look for unified representations for other incomplete networks such as the incomplete star networks and the incomplete rotator graphs, and then to develop broadcasting algorithms for them with the aid of these representations.

REFERENCES

1. S. B. Akers and B. Krishnamurthy, A group-theoretic model for symmetry interconnection networks, *IEEE Trans. Comput.* **38**, No. 4 (1989), 555–566.
2. N. S. Bowen, C. N. Nikolaou, and A. Ghafoor, On the assignment problem of arbitrary process systems to heterogeneous distributed computer systems, *IEEE Trans. Comput.* **41**, No. 3 (1992), 257–273.
3. F. Buckley and F. Harary, "Distance in Graph," Addison–Wesley, Reading, MA, 1990.
4. G. H. Chen and D. R. Duh, Topological properties, communication, and computation on WK-recursive networks, *Networks* **24**, No. 6 (1994), 303–317.
5. P. Corbett, Rotator graphs: An efficient topology for point-to-point multiprocessor networks, *IEEE Trans. Parallel Distrib. Sytems* **3**, No. 5 (1992), 622–626.
6. D. R. Duh and G. H. Chen, Topological properties of WK-recursive networks, *J. Parallel Distrib. Comput.* **23**, No. 3 (1994), 468–474.
7. D. R. Duh and G. H. Chen, On the Rabin number problem, *Networks* **30**, No. 3 (1997), 219–230.
8. R. Fernandes, Recursive interconnection networks for multicomputer networks, *in* "Proceedings of the International Conference on Parallel Processing, Vol. I, 1992," pp. 76–79.
9. R. Fernandes and A. Kanevsky, Hierarchical WK-recursive topologies for multicomputer systems, *in* "Proceedings of the International Conference on Parallel Processing, Vol. I, 1993," pp. 315–318.
10. R. Fernandes and A. Kanevsky, Substructure allocation in recursive interconnection networks, *in* "Proceedings of International Conference on Parallel Processing, Vol. I, 1993," pp. 319–322.
11. R. Fernandes, D. K. Griesen, and A. Kanevsky, Efficient routing and broadcasting in recursive interconnection networks, *in* "Proceeding of the International Conference on Parallel Processing, 1994," pp. 51–58.
12. R. Fernandes, D. K. Friesen, and A. Kanevsky, Embedding rings in recursive networks, *in* "Proceedings of International Symposium on Parallel and Distributed Processing, Oct. 1994," pp. 273–280.
13. K. Hwang and J. Ghosh, Hypernet: A communication efficient architecture for constructing massively parallel computers, *IEEE Trans. Comput.* **36**, No. 12 (1987), 1450–1466.
14. A. Iazzetta, C. Sanges, and U. Scafuri, Special multicast wormhole routing for WK-networks, manuscript.
15. K. B. Irani and K. Chen, Minimization of interprocessor communication for parallel computation, *IEEE Trans. Comput.* **C-31** (1982), 1067–1075.
16. S. Johnsson and C. T. Ho, Optimal broadcasting and personalized communication in hypercubes, *IEEE Trans. Comput.* **38**, No. 9 (1989), 1249–1268.
17. H. P. Katseff, Incomplete hypercubes, *IEEE Trans. Comput.* **C-37**, No. 5 (1988), 604–608.
18. S. Latifi and N. Bagherzadeh, Incomplete star: An incrementally scalable network based on the star graph, *IEEE Trans. Parallel Distrib. Systems* **5**, No. 1 (1994), 97–102.
19. S. Latifi and N. Bagherzadeh, On clustered-star graph and its properties, manuscript.
20. D. T. Lee and F. P. Preparata, Computational geometry—a survey, *IEEE Trans. Comput.* **C-33** (1984), 1072–1101.
21. S. Lee and K. G. Shin, Interleaved all-to-all reliable broadcast on meshes and hypercubes, *IEEE Trans. Parallel Distrib. Systems* **5**, No. 5 (1994), 449–458.

22. V. E. Mendia and D. Sarkar, Optimal broadcasting on the star graph, *IEEE Trans. Parallel Distrib. Systems* **3**, No. 4 (1992), 389–396.

23. S. Ponnuswamy and V. Chaudhary, Embedding of cycles in rotator and incomplete rotator graphs, *in* "Proceedings of International Symposium on Parallel and Distributed Processing, Oct. 1994," pp. 603–610.

24. F. P. Preparata and J. Vuillemin, The cube-connected cycles: A versatile network for parallel computation, *Comm. ACM* **24**, No. 5 (1981), 300–309.

25. C. P. Ravikumar, A. Kuchlous, and G. Manimaran, Incomplete star graph: An economical fault-tolerant interconnection network, *in* "Proceedings of International Conference on Parallel Processing, Vol. I, 1993," pp. 83–90.

26. M. R. Samatham and D. K. Pradhan, The de Bruijn multiprocessor networks: A versatile parallel processing and sorting networks for VLSI, *IEEE Trans. Comput.* **38**, No. 4 (1989), 567–581.

27. A. Sen, Supercube: An optimally fault tolerant network architecture, *Acta Inform.* **26** (1989), 741–748.

28. J. P. Sheu, C. T. Wu, and T. S. Chen, An optimal broadcasting algorithm without message redundancy in star graphs, *IEEE Trans. Parallel Distrib. Systems* **6**, No. 6 (1995), 653–658.

29. H. S. Stone, Parallel processing with the perfect shuffle, *IEEE Trans. Comput.* **20**, No. 2 (1971), 153–161.

30. M. Y. Su, G. H. Chen, and D. R. Duh, A shortest-path routing algorithm for the incomplete WK-recursive networks, *IEEE Trans. Parallel Distrib. Systems* **8**, No. 4 (1997), 367–379.

31. M. Y. Su, G. H. Chen, and D. R. Duh, Topological properties of incomplete WK-recursive networks, *Networks* **32**, No. 1 (1998), 23–45.

32. M. Y. Su, G. H. Chen, and D. R. Duh, "A Broadcasting Algorithm for Incomplete WK-Recursive Networks," Technical Report 95-09, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, December 1995.

33. N. F. Tzeng, An effective approach to the enhancement of incomplete hypercube computers, *J. Parallel Distrib. Comput.* **14** (1992), 163–174.

34. G. D. Vecchia and C. Sanges, A recursively scalable network VLSI implementation, *Future Generation Computer Systems* **4**, No. 3 (1988), 235–243.

35. G. D. Vecchia and C. Sanges, An optimized broadcasting technique for WK-recursive topologies, *Future Generation Computer Systems* **4**, No. 3 (1989/1990), 353–357.

MING-YANG SU received the B.S. degree from Tunghai University in June 1989, the M.S. degree from National Central University in June 1991, and the Ph.D. degree from National Taiwan University in January 1997, all in computer science. He is currently an associate professor in the Department of Electronic Engineering, Hwa Hsia College of Technology and Commerce, Taipei, Taiwan. His research interests include parallel and distributed systems, graph theory, and fault-tolerant computing.

GEN-HUEY CHEN was born in Taiwan in 1959. He received the B.S. degree in computer science from National Taiwan University in June 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Taiwan in June 1983 and January 1987, respectively. He joined the faculty of the Department of Computer Science and Information Engineering, National Taiwan University in February 1987, and he has been a professor since August 1992. He received the Distinguished Research Award from the National Science Council, Taiwan, in 1993, 1995, and 1997. His current research interests include graph-theoretic interconnection networks, parallel and distributed computing, and design and analysis of algorithms.

DYI-RONG DUH received the B.S. degree in electronic engineering from National Taiwan Institute of Technology, the M.S. degree in computer engineering from Tamkang University, and the Ph.D. degree in computer science from National Taiwan University in 1983, 1988, and 1994, respectively. He is currently an associate professor in the Department of Computer Science and Information Engineering, National Chi Nan University. His research interests include graph-theoretic interconnection networks, fault-tolerant computing, and computer architectures.