

Broadcasting on Incomplete WK-Recursive Networks

Min-Yang Su and Gen-Huey Chen

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

Dyi-Rong Duh

Department of Electronic Engineering,
Hwa Shia Junior College, Taipei, TAIWAN

Abstract

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, experimental results show that the heights of the broadcasting trees do not exceed the diameters, and a high percentage of the nodes can receive the message from the source node via the shortest paths.

1 Introduction

In the recent decade, a number of networks have been announced in the literature. 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.

Although the WK-recursive networks own many favorable properties (see [2, 3, 4-6, 7, 16, 17]), 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 $t \geq 1$ is the level of expansion. Thus, as $d=4$, extra $3 \cdot 4^7 = 49152$ 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, such as incomplete hypercubes [8], incomplete stars [9, 12], incomplete rotators [11] and incomplete WK-recursive networks [13] have been defined recently.

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 have been 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 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. The incomplete WK-recursive networks are then defined as their induced subgraphs (graphs and networks are used interchangeably in this paper). It is also shown that the incomplete WK-recursive networks can be conveniently expressed as multistage graphs.

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 $t \geq 1$. $K(d, 1)$, which is the basic building block, is a d -node complete graph, and $K(d, t)$ for $t \geq 2$ 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.

Definition 2.1.[2] The node set of $K(d, t)$ is denoted by $\{a_{t-1}a_{t-2}\dots a_1a_0 \mid a_i \in \{0, 1, \dots, d-1\} \text{ for } 0 \leq i \leq t-1\}$. Node adjacency is defined as follows: $a_{t-1}a_{t-2}\dots a_1a_0$ is adjacent to (1) $a_{t-1}a_{t-2}\dots a_1b$, where $0 \leq b \leq d-1$ and $b \neq a_0$, and (2) $a_{t-1}a_{t-2}\dots a_{j+1}a_{j-1}(a_j)^j$ if $a_j \neq a_{j-1}$ and $a_{j-1} = a_{j-2} = \dots = a_0$, where $1 \leq j \leq t-1$ and $(a_j)^j$ represents j consecutive a_j 's. The links of (1) are called *substituting links*, and are labeled 0. The link of (2), if existing, is called *j-flipping link* (or simply *flipping link*), and is labeled j . Besides, if $a_{t-1} = a_{t-2} = \dots = a_0$, there is a link, called *open link*, incident to node $a_{t-1}a_{t-2}\dots a_1a_0$. The open link is labeled t . Since the open link is reserved for further expansion, its other end node is unspecified.

$K(d, t)$ 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 Figure 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.

Definition 2.2. Define $c_{t-1}c_{t-2}\dots c_m \cdot K(d, m)$ to be the induced subgraph of $K(d, t)$ by $\{c_{t-1}c_{t-2}\dots c_m a_{m-1}\dots a_1a_0 \mid a_j \in \{0, 1, \dots, d-1\} \text{ for } 0 \leq j \leq m-1\}$, where $1 \leq m < t$ and $c_{t-1}, c_{t-2}, \dots, c_m$ are all integers from $\{0, 1, \dots, d-1\}$.

For example, refer to Figure 1, where $31 \cdot K(4, 1)$ is the induced subgraph of $K(4, 3)$ by $\{310, 311, 312, 313\}$.

Definition 2.3. Node $a_{t-1}a_{t-2}\dots a_1a_0$ is a k -frontier if $a_{k-1} = \dots = a_1 = a_0$, where $1 \leq k \leq t$.

By Definition 2.3, a k -frontier is automatically an l -frontier for $1 \leq l < k$. 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.

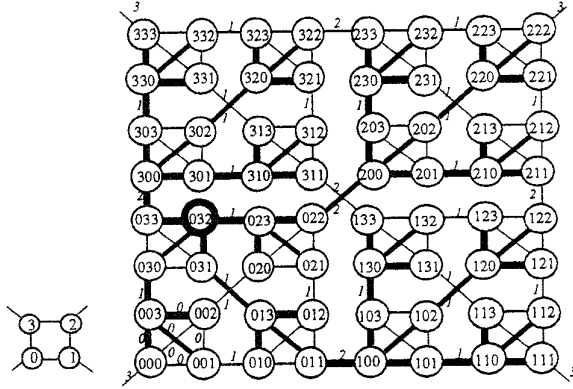


Figure 1. The structures of $K(4, 1)$ and $K(4, 3)$. The latter also shows the spanning tree rooted at 032 that results from Chen and Duh's broadcasting algorithm.

The incomplete WK-recursive networks [13] 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 .

The coefficient vector of $IK(d, t)$ is uniquely defined as a $(t-1)$ -vector $(b_{t-1}, b_{t-2}, \dots, b_1)$ such that $N = b_{t-1}d^{t-1} + b_{t-2}d^{t-2} + \dots + b_1d$, where $0 \leq b_m \leq d-1$ for all $1 \leq m \leq t-1$. $IK(d, t)$ with coefficient vector $(b_{t-1}, b_{t-2}, \dots, b_1)$ contains b_m embedded $K(d, m)$'s, i.e., $b_{t-1}b_{t-2} \dots b_{m+1}0 \cdot K(d, m)$, $b_{t-1}b_{t-2} \dots b_{m+1}1 \cdot K(d, m)$, ..., and $b_{t-1}b_{t-2} \dots b_{m+1}(b_m-1) \cdot K(d, m)$, where $1 \leq m \leq t-1$. For example, $IK(5, 6)$ with coefficient vector $(2, 3, 0, 4, 2)$ contains the following embedded $K(d, m)$'s.

- $0 \cdot K(5, 5)$, $1 \cdot K(5, 5)$
- $20 \cdot K(5, 4)$, $21 \cdot K(5, 4)$, $22 \cdot K(5, 4)$
- $2300 \cdot K(5, 2)$, $2301 \cdot K(5, 2)$, $2302 \cdot K(5, 2)$, $2303 \cdot K(5, 2)$
- $23040 \cdot K(5, 1)$, $23041 \cdot K(5, 1)$

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

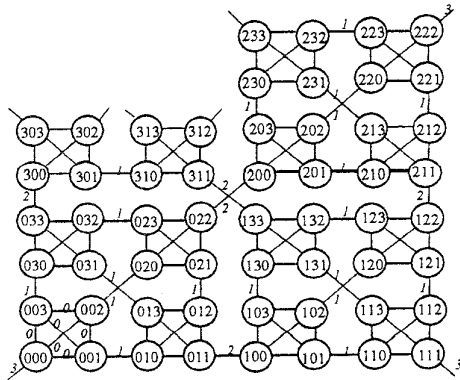


Figure 2. The structures of $IK(4, 3)$ with coefficient vector $(3, 2)$.

Let S_m represent the subgraph of $IK(d, t)$ with coefficient vector $(b_{t-1}, b_{t-2}, \dots, b_i, *)$ induced by the nodes of $b_{t-1}b_{t-2} \dots b_{m+1}0 \cdot K(d, m)$, $b_{t-1}b_{t-2} \dots b_{m+1}1 \cdot K(d, m)$, ..., and $b_{t-1}b_{t-2} \dots b_{m+1}(b_m-1) \cdot K(d, m)$, where $i \leq m \leq t-1$. For example, when $IK(5, 6)$ with coefficient vector $(2, 3, 0, 4, 2, *)$ is concerned, S_5 contains $0 \cdot K(5, 5)$ and $1 \cdot K(5, 5)$, and S_2 contains $2300 \cdot K(5, 2)$, $2301 \cdot K(5, 2)$, $2302 \cdot K(5, 2)$ and $2303 \cdot K(5, 2)$. We note that there is an m -flipping link between any two of these b_m embedded $K(d, m)$'s within S_m .

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

There are $\min\{b_m, b_{m-1}\}$ m -flipping links between S_m and S_{m-1} , each connecting C_m^j and C_{m-1}^j for some $0 \leq j \leq \min\{b_m, b_{m-1}\}-1$. Besides, for $t-1 \geq u > v \geq i$ and $u-v > 1$, there may exist a u -flipping link between S_u and S_v . If such a link exists, it is called a *jumping u -flipping link*. A necessary and sufficient condition for the existence of jumping flipping links has been suggested in [14] as follows.

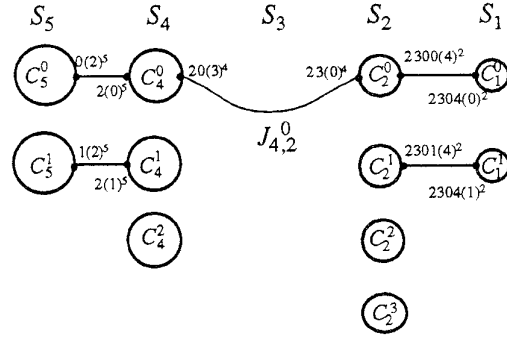


Figure 3. The multistage graph representation of $IK(6, 5)$ with coefficient vector $(2, 3, 0, 4, 2, *)$.

Theorem 2.1.[14] For $IK(d, t)$ with coefficient vector $(b_{t-1}, b_{t-2}, \dots, b_i, *)$, one jumping u -flipping link exists between S_u and S_v if and only if $b_u > b_{u-1} = b_{u-2} = \dots = b_{v+1} < b_v$, where $t-1 \geq u > v \geq i$ and $u-v > 1$. Moreover, this jumping flipping link connects C_u^e and C_v^e , where $e = b_{u-1} = b_{u-2} = \dots = b_{v+1}$.

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 Figure 3 for illustration).

The structure of $S_{t-1} + S_{t-2} + \dots + S_i$ is further detailed as follows. Since each C_m^j , where $t-1 \geq m \geq i$ and $0 \leq j \leq b_m-1$, 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 $C_m^0, C_m^1, \dots, C_m^{j-1}, C_m^{j+1}, \dots,$ and $C_m^{b_m-1}$, respectively; (2) one m -flipping link connecting C_{m-1}^j if $j \leq b_{m-1}-1$, or one jumping m -flipping link connecting C_l^j , where $l < m-1$, if $j = b_{m-1} = b_{m-2} = \dots = b_{l+1} < b_l$; (3) one $(m+1)$ -flipping link connecting C_{m+1}^j if $j \leq b_{m+1}-1$, or one jumping h -flipping link connecting C_h^j , where $h > m+1$, if $b_h > b_{h-1} = b_{h-2} = \dots = b_{m+1} = j$. Both end nodes of (1) are $b_{l-1}b_{l-2} \dots b_{m+1}j(x)^m \in C_m^j$ and $b_{l-1}b_{l-2} \dots b_{m+1}x(j)^m \in C_m^x$, where $0 \leq x \leq b_m-1$ and $x \neq j$. Both end nodes of (2) are $b_{l-1}b_{l-2} \dots b_{m+1}j(b_m)^m \in C_m^j$ and $b_{l-1}b_{l-2} \dots b_{m+1}j(b_m)^m \in C_{m-1}^j$ (or $\in C_l^j$). Both end nodes of (3) are $b_{l-1}b_{l-2} \dots b_{m+2}j(b_{m+1})^{m+1} \in C_{m+1}^j$ (or $b_{l-1}b_{l-2} \dots b_{h+1}j(b_h)^h \in C_h^j$) and $b_{l-1}b_{l-2} \dots b_{m+2}j(b_{m+1})^{m+1} \in C_m^j$.

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 $(b_{t-1}, b_{t-2}, \dots, b_i, *)$, where $t-1 \geq z \geq i$. 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 $S_m + S_{m-1} + \dots + S_n$, where $t-1 \geq m \geq z \geq n \geq i$. 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 [2] 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 following 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, Figure 1 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 has been proved in [2].

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

With slight modification, Chen and Duh's algorithm can disseminate a message within any embedded $K(d, l)$, where $1 \leq l < t$. We assume $r = r_{l-1}r_{l-2} \dots r_1r_0$. A spanning tree of $r_{l-1}r_{l-2} \dots r_{l+1}r_lK(d, l)$ rooted at node r can be obtained by executing Chen and Duh's algorithm, provided step 3 is modified as follows.

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 $K(d, l)$. As a consequence of Lemma 3.1, the resulting spanning tree has height at most 2^{l-1} .

Now we are going to construct $ST(m, n, r, m)$ in $IK(d, t)$ ($=S_{t-1}+S_{t-2}+\dots+S_i$), where $t-1 \geq m \geq n \geq i$. Recall that S_m contains b_m embedded $K(d, m)$'s that are completely connected by m -flipping links. Moreover, since $r \in S_m$, we have $r = r_{t-1}r_{t-2} \dots r_{m+1}r_m \dots r_1r_0 = b_{t-1}b_{t-2} \dots b_{m+1}r_m \dots r_1r_0 \in b_{t-1}b_{t-2} \dots b_{m+1}r_m \cdot K(d, m)$, where $0 \leq r_m \leq b_m-1$. 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 following components.

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

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

Pattern A.

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

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

Pattern B.

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

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

Pattern C.

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

- $ST(m, m, r, m)$.
- Link set $\{(b_{t-1}b_{t-2} \dots b_{m+1}x(b_m)^m, b_{t-1}b_{t-2} \dots b_{m+1}b_m(x)^m) \mid 0 \leq x \leq b_{m-1}-1\}$.
- Spanning trees of $b_{t-1}b_{t-2} \dots b_{m+1}b_mx \cdot K(d, m-1)$ rooted at $b_{t-1}b_{t-2} \dots b_{m+1}b_mx(r_m)^m$ for all $0 \leq x \leq b_{m-1}-1$.

- Jumping flipping link $(b_{i-1}b_{i-2}\dots b_{m+1}b_{m-1}(b_m)^m, b_{i-1}b_{i-2}\dots b_{m+1}b_m(b_{m-1})^m)$.
- $ST(m-2, n, r', l)$, where $r'=b_{i-1}b_{i-2}\dots b_{m+1}b_m(b_{m-1})^m=b_{i-1}b_{i-2}\dots b_{l+1}(b_{m-1})^{l+1}$.

Pattern D.

If $r_m \geq b_{m-1}$ and there is no jumping flipping link from S_m to some S_i , where $m-1 > l \geq n$, then we determine the leftmost jumping flipping link, say $J_{u,v}^e$, in $S_m+S_{m-1}+\dots+S_n$, if it exists.

Since $J_{u,v}^e$ is the leftmost one, by Theorem 2.1 we have $b_m > (r_m \geq) b_{m-1} \geq b_{m-2} \geq \dots \geq b_u > b_{u-1} = b_{u-2} = \dots = b_{v+1} < b_v$ and $e = b_{u-1} = b_{u-2} = \dots = b_{v+1}$. Then $ST(m, n, r, m)$ is constructed as the union of the following components (refer to Figure 4(d)).

- $ST(m, m, r, m)$.
- Link sets $\{(b_{i-1}b_{i-2}\dots b_{j+1}x(b_j)^j, b_{i-1}b_{i-2}\dots b_{j+1}b_j(x)^j) \mid 0 \leq x \leq b_{j-1}-1\}$ for all $u \leq j \leq m$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{j+1}b_jx \cdot K(d, j-1)$ rooted at $b_{i-1}b_{i-2}\dots b_{j+1}b_j(x)^j$ for all $0 \leq x \leq b_{j-1}-1$ and all $u \leq j \leq m$.
- Jumping flipping link $(b_{i-1}b_{i-2}\dots b_{u+1}e(b_u)^u, b_{i-1}b_{i-2}\dots b_{u+1}b_u(e)^u)$.
- $ST(u-2, n, r', v)$, where $r'=b_{i-1}b_{i-2}\dots b_{u+1}b_u(e)^u=b_{i-1}b_{i-2}\dots b_{v+1}(e)^{v+1}$.

If $J_{u,v}^e$ 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 $b_{m'}=b_{m'-1}=\dots=b_{k+1}<b_k$. If $m'=k$, $ST(m', n, r', k)$ is constructed recursively. Otherwise, $ST(m', n, r', k)$ is constructed as follows.

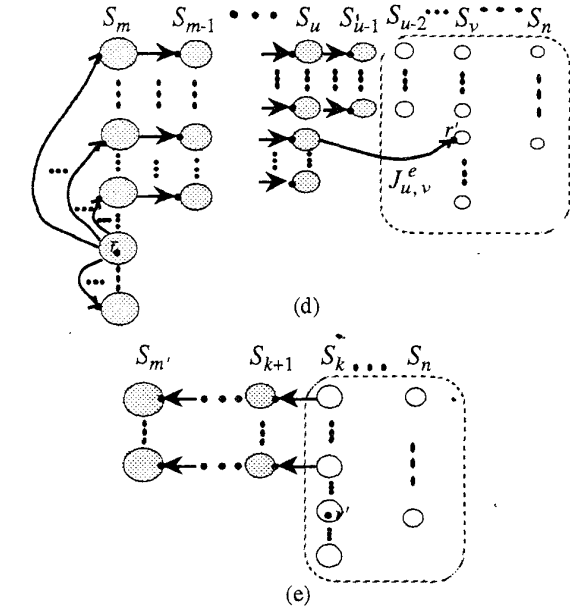
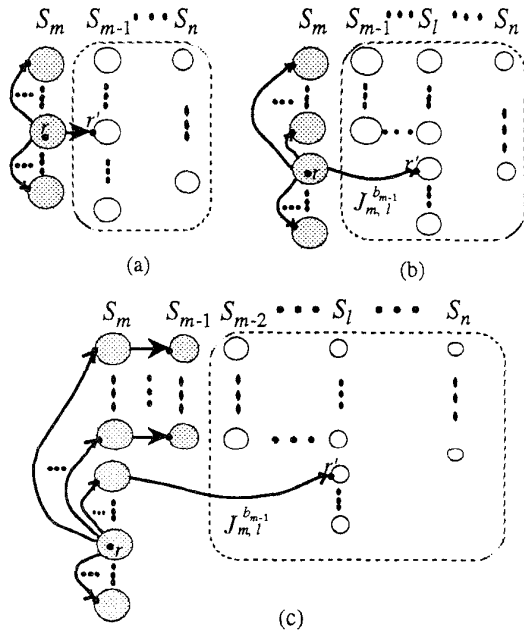


Figure 4. Five dissemination patterns for constructing $ST(m, n, r, m)$. (a) **Pattern A**. (b) **Pattern B**. (c) **Pattern C**. (d) **Pattern D**. (e) **Pattern E**.

Pattern E.

$ST(m', n, r', k)$, where $b_{m'}=b_{m'-1}=\dots=b_{k+1}<b_k$ and $m'>k \geq n$, is constructed as the union of the following components (assuming $h=b_{m'}=b_{m'-1}=\dots=b_{k+1}$) (refer to Figure 4(e)).

- $ST(k, n, r', k)$.
- Link sets $\{(b_{i-1}b_{i-2}\dots b_{j+1}b_j(x)^j, b_{i-1}b_{i-2}\dots b_{j+1}x(b_j)^j) \mid 0 \leq x \leq h-1\}$ for all $k < j \leq m'$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{j+1}x \cdot K(d, j)$ rooted at $b_{i-1}b_{i-2}\dots b_{j+1}x(b_j)^j$ for all $0 \leq x \leq h-1$ and all $k < j \leq m'$.

With these patterns, $ST(m, n, r, m)$, where $m > n$, can be constructed recursively. For example, let us consider $IK(d, 11)$ with $d > 5$ and coefficient vector $(4, 5, 4, 2, 2, 4, 2, 2, 1, 5, *)$. Assuming $r \in 453 \cdot K(d, 8)$, $ST(8, 2, r, 8)$ can be constructed as follows (refer to Figure 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**.

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 $r=r_{i-1}r_{i-2}\dots r_1r_0$ 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 $n \leq j < m$, will grow toward the left when the patterns are applied. The details are shown as follows.

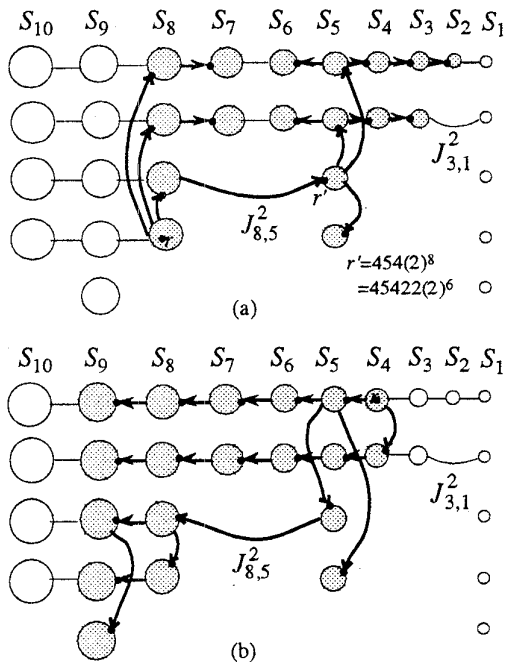


Figure 5. Examples. (a) $ST(8, 2, r, 8)$, where r belongs to $453.K(d, 8)$. (b) $ST(9, 4, r, 4)$, where r belongs to $4542240.K(d, 4)$

Pattern F.

If $b_j \leq b_{j+1}$, then $ST(j, n, r, n)$ grows into $ST(j+1, n, r, n)$ by augmenting the following components (refer to Figure 6(a)).

- Link set $\{(b_{i-1}b_{i-2}\dots b_{j+2}b_{j+1}(x)^{j+1}, b_{i-1}b_{i-2}\dots b_{j+2}x(b_{j+1})^{j+1}) \mid 0 \leq x \leq b_j - 1\}$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{j+2}x \cdot K(d, j+1)$ rooted at $b_{i-1}b_{i-2}\dots b_{j+2}x(b_{j+1})^{j+1}$ for all $0 \leq x \leq b_j - 1$.
- Link set $\{(b_{i-1}b_{i-2}\dots b_{j+2}w(y)^{j+1}, b_{i-1}b_{i-2}\dots b_{j+2}y(w)^{j+1}) \mid b_j \leq y \leq b_{j+1} - 1\}$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{j+2}y \cdot K(d, j+1)$ rooted at $b_{i-1}b_{i-2}\dots b_{j+2}y(w)^{j+1}$ for all $b_j \leq y \leq b_{j+1} - 1$.

The subnetwork $b_{i-1}b_{i-2}\dots b_{j+2}w \cdot K(d, j+1)$ is responsible for disseminating the message to $b_{i-1}b_{i-2}\dots b_{j+2}y \cdot K(d, j+1)$ for all $b_j \leq y \leq b_{j+1} - 1$. 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 $m \geq l > j+1$, then $ST(j, n, r, n)$ grows into $ST(j+1, n, r, n)$ by augmenting the following components (refer to Figure 6(b)).

- Link set $\{(b_{i-1}b_{i-2}\dots b_{j+2}b_{j+1}(x)^{j+1}, b_{i-1}b_{i-2}\dots b_{j+2}x(b_{j+1})^{j+1}) \mid 0 \leq x \leq b_{j+1} - 1\}$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{j+2}x \cdot K(d, j+1)$ rooted at $b_{i-1}b_{i-2}\dots b_{j+2}x(b_{j+1})^{j+1}$ for all $0 \leq x \leq b_{j+1} - 1$.

Pattern H.

If $b_j > b_{j+1}$ and there exists one jumping flipping link from S_j to some S_l , where $m \geq l > j+1$, then $ST(j, n, r, n)$ grows into $ST(l, n, r, n)$ by augmenting the following components (refer to Figure 6(c)).

n, r, n by augmenting the following components (refer to Figure 6(c)), where $e = b_{l-1} = b_{l-2} = \dots = b_{j+1}$ is assumed.

- Link sets $\{(b_{i-1}b_{i-2}\dots b_{s+1}b_s(x)^s, b_{i-1}b_{i-2}\dots b_{s+1}x(b_s)^s) \mid 0 \leq x \leq e - 1\}$ for all $j < s \leq l$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{s+1}x \cdot K(d, s)$ rooted at $b_{i-1}b_{i-2}\dots b_{s+1}x(b_s)^s$ for all $0 \leq x \leq e - 1$ and all $j < s \leq l$.
- Jumping flipping link $(b_{i-1}b_{i-2}\dots b_{l+1}b_l(e)^l, b_{i-1}b_{i-2}\dots b_{l+1}e(b_l)^l)$.
- Spanning tree of $b_{i-1}b_{i-2}\dots b_{l+1}e \cdot K(d, l)$ rooted at $b_{i-1}b_{i-2}\dots b_{l+1}e(b_l)^l$.
- Link set $\{(b_{i-1}b_{i-2}\dots b_{l+1}e(y)^l, b_{i-1}b_{i-2}\dots b_{l+1}y(e)^l) \mid e + 1 \leq y \leq b_{l-1} - 1\}$.
- Spanning trees of $b_{i-1}b_{i-2}\dots b_{l+1}y \cdot K(d, l)$ rooted at $b_{i-1}b_{i-2}\dots b_{l+1}y(e)^l$ for all $e + 1 \leq y \leq b_{l-1} - 1$.

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.

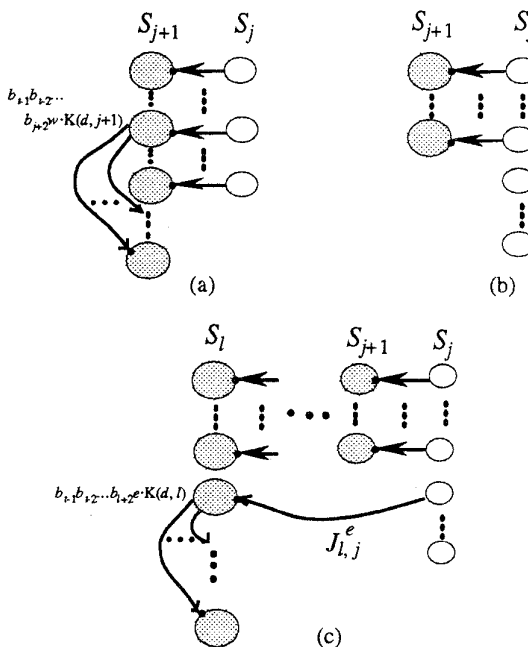


Figure 6. Three dissemination patterns for constructing $ST(m, n, r, n)$. (a) **Pattern F.** (b) **Pattern G.** (c) **Pattern H.**

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 Figure 5(b) where $r \in 4542240.K(d, 4)$ and $ST(9, 4, r, 4)$ is shown. The $IK(d, t)$ 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 $IK(d, t)$ by constructing $ST(m, n, r, z)$, where $m \leq z \leq n$. 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 $r \in b_{t-1}b_{t-2} \dots b_{z+1}\alpha \cdot K(d, z)$, where $0 \leq \alpha \leq b_{z-1}$. 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 $J_{u,v}^e$ exists such that $m \geq u > z > v \geq n$. 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 $J_{u,v}^e$ exists, by Theorem 2.1 we have $b_u > b_{u-1} = b_{u-2} = \dots = b_{v+1} < b_v$ and $e = b_{u-1} = b_{u-2} = \dots = b_{v+1}$. $ST(m, n, r, z)$ is constructed according to the following two cases.

Case 1. $z = u - 1$.

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

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

The subnetwork $b_{t-1}b_{t-2} \dots b_{u+1}e \cdot K(d, u)$ in S_u receives the message via $J_{u,v}^e$ 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 $S_m + S_{m-1} + \dots + S_{u+1}$, 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 following components (refer to Figure 7(b) where $\alpha = e - 1$ is assumed).

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

Like Case 1, $ST(u, n, r, z)$ then grows into $ST(m, n, r, z)$ by augmenting the nodes of $S_m + S_{m-1} + \dots + S_{u+1}$ 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 [15].

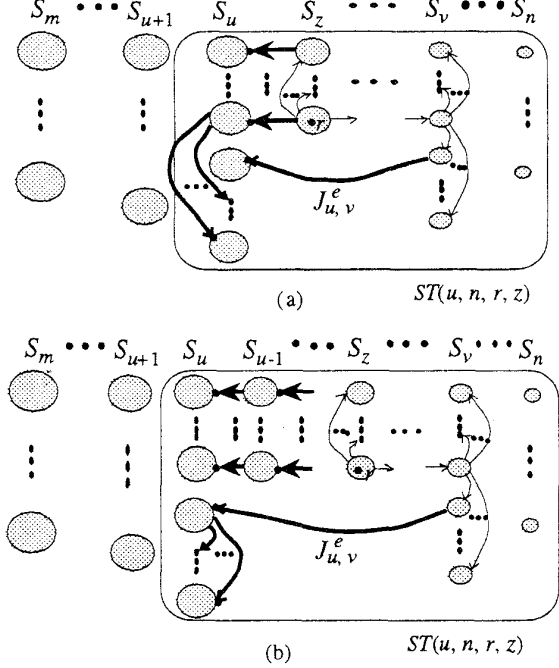


Figure 7. Illustration of $ST(m, n, r, z)$. (a) $z = u - 1$. (b) $z < u - 1$.

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. The following 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 $IK(d, t)$ [13].

Remark. Using the prune-and-search technique [10], the algorithm can compute the diameter of $IK(d, t)$ in $O(t)$ time. Moreover, the farthest pair of nodes can be determined simultaneously. Note that although the diameter of $IK(d, t)$ can be computed, no formula is available for computing it.

- Su, Chen, and Duh's shortest-path routing algorithm for $IK(d, t)$ [14].
- Chen and Duh's shortest-path routing algorithm for $K(d, t)$ [2].
- Chen and Duh's broadcasting algorithm for $K(d, t)$ [2] (also reviewed in the beginning of Section 3).

We first compared the maximum transmission length, i.e., the height of $ST(t-1, i, r, z)$, with the diameter of $IK(d, t)$. Table I shows the experimental result. For each of the entries marked with *, the experiment was made exhaustively. That is, for each $d^{l-1} < N < d^l$, 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 [13].

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* [1] if the distance between it and its farthest node is equal to the diameter of the network. Su *et al.*'s algorithm [13] 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 [2] for $K(d, t)$ was adopted in the experiment. The experiment proceeded as follows. First, 1000 instances were randomly chosen for both $IK(d, t)$ and $K(d, t)$. Here a selected value of N combined with a selected source node r forms an instance of $IK(d, t)$, whereas a selected source node alone forms an instance of $K(d, t)$. For each of the chosen instances, the percentage was computed by the aid of Su *et al.*'s algorithm [14] and Chen and Duh's algorithm [2]. The two algorithms can compute the distance of arbitrary two nodes in $IK(d, t)$ and $K(d, t)$, respectively. The average percentages for $IK(d, t)$ and $K(d, t)$ were then computed for the 1000 instances (we found in the experiment that all average percentages got stable after running 1000 instances).

Table I. Experimental results about maximum transmission length.

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

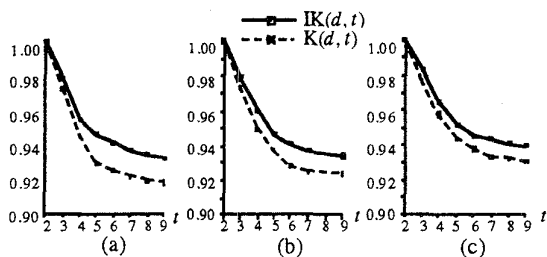


Figure 8. Average percentages of nodes that receive the message via the shortest paths. (a) $d=4$. (b) $d=5$. (c) $d=6$.

7 Concluding Remarks

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 the maximum transmission length does not exceed the diameter, and a high percentage of nodes can receive the message from the source node via the shortest paths.

REFERENCES

- [1] F. Buckley and F. Harary, *Distance in Graph*, Addison-Wesley, 1990.
- [2] G. H. Chen and D. R. Duh, "Topological properties, communication, and computation on WK-recursive networks," *Networks*, vol. 24, no. 6, pp. 303-317, 1994.
- [3] D. R. Duh and G. H. Chen, "Topological properties of WK-recursive networks," *J. of Parallel and Distributed Computing*, vol. 23, no. 3, pp. 468-474, 1994.
- [4] R. Fernandes and A. Kanevsky, "Substructure allocation in recursive interconnection networks," in *Proc. of Int. Conf. on Parallel Processing*, vol. I, 1993, pp. 319-322.
- [5] R. Fernandes, D. K. Griesen, and A. Kanevsky, "Efficient routing and broadcasting in recursive interconnection networks," in *Proc. of the Int. Conf. on Parallel Processing*, 1994, pp. 51-58.
- [6] R. Fernandes, D. K. Griesen, and A. Kanevsky, "Embedding rings in recursive networks," in *Proc. of Int. Symp. on Parallel and Distributed Processing*, 1994, pp. 273-280.
- [7] A. Iazzetta, C. Sanges, and U. Scafuri, "Special multicast wormhole routing for WK-networks," manuscript.
- [8] H. P. Katseff, "Incomplete hypercubes," *IEEE Trans. on Computers*, vol. C-37, no. 5, pp. 604-608, 1988.
- [9] S. Latifi and N. Bagherzadeh, "Incomplete star: an incrementally scalable network based on the star graph," *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 1, pp. 97-102, 1994.
- [10] D. T. Lee and F. P. Preparata, "Computational geometry - a survey," *IEEE T. C.*, vol. C-33, pp. 1072-1101, Dec. 1984.
- [11] S. Ponnuswamy and V. Chaudhary, "Embedding of cycles in rotator and incomplete rotator graphs," in *Proc. of Int. Symp. on Parallel and Distributed Processing*, Oct. 1994, pp. 603-610.
- [12] C. P. Ravikumar, A. Kuchlous, and G. Manimaran, "Incomplete star graph: an economical fault-tolerant interconnection network," in *Proc. of Int. Conf. on Parallel Processing*, vol. I, 1993, pp. 83-90.
- [13] M. Y. Su, G. H. Chen, and D. R. Duh, "Topological properties of incomplete WK-recursive networks," Tech. Rep. NTUCSIE 95-06, National Taiwan Univ., Taipei, Taiwan, March 1995.
- [14] M. Y. Su, G. H. Chen, and D. R. Duh, "A shortest-path routing algorithm for the incomplete WK-recursive networks," Tech. Rep. NTUCSIE 95-07, National Taiwan Univ., Taipei, Taiwan, July 1995.
- [15] M. Y. Su, G. H. Chen, and D. R. Duh, "A broadcasting algorithm for incomplete WK-recursive networks," Tech. Rep. NTUCSIE 95-09, National Taiwan Univ., Taipei, Taiwan, December 1995.
- [16] G. D. Vecchia and C. Sanges, "A recursively scalable network VLSI implementation," *Future Generation Computer Systems*, vol. 4, no. 3, pp. 235-243, 1988.
- [17] G. D. Vecchia and C. Sanges, "An optimized broadcasting technique for WK-recursive topologies," *Future Generation Computer Systems*, vol. 4, no. 3, pp. 353-357, 1989/90.