Novel Broadcasting Algorithm of the Complete Recursive Network

Jywe-Fei Fang
National Taichung University, Department of Digital Content and Technology
140 Ming-Shen Rd., Taichung 403, Taiwan, R. O. C.
E-mail1: jffang@mail.ntcu.edu.tw

Wen-Yew Liang
National Taipei University of Technology, Department of Computer Science and Information Engineering

Hong-Ren Chen
National Taichung University, Department of Digital Content and Technology

Ka-Lok Ng
Asia University, Department of Biotechnology and Bioinformatics

Keywords: complete WK-Recursive networks, interconnection networks, broadcasting algorithms

Received: November 20, 2005

The interconnection network considered in this paper is the complete WK-Recursive network that demonstrates many attractive properties, such as high degree of regularity, symmetry and efficient communication. Chen and Duh have proposed a distributed stack-base broadcasting algorithm for the complete WK-Recursive networks [Networks, 24 (1994) 303-317]. To perform this algorithm, a stack of O(log N) elements, where N is the number of nodes, to keep the labels of links is included in each message. Moreover, as a node receives the message, a series of O(log N) pop and push operations on the stack is required. In this paper, we present a novel broadcasting algorithm for the complete WK-Recursive network, which is much simpler and requires only constant data included in each message and constant time to determine the neighbors to forward the message.

Povzetek: Opisan je nov algoritem razpošiljanja v rekurzivni mreži.

1 Introduction

In massively parallel MIMD systems, the topology plays a crucial role in issues such as communication performance, hardware cost, potentialities for efficient applications and fault tolerance capabilities. A topology named *complete WK-Recursive network* has been proposed by Vecchia and Sanges under CAPRI(Concurrent Architecture and Programming environment for highly Integrated systems) project supported by the Strategic Program on Parallel Computing of the National Research Council of Italy [11]. A complete WK-Recursive network with amplitude W and level L is denoted by WK(W, L), where $W \geq 2$. The topology has many attractive properties, such as high degree of regularity, symmetry and efficient communication. Particularly, for any specified number of degree, it can be expanded to arbitrary size level without reconfiguring the links. The complete WK-Recursive networks have received considerable attention. Researchers have devoted themselves to various issues of complete WK-Recursive networks. A VLSI implementation and a simple routing algorithm of complete WK-Recursive networks have been developed [11]. Verdoscia and Vaccaro proposed an adaptive routing algorithm on the complete WK-Recursive networks [12]. The topological properties of complete WK-Recursive networks are studied [7]. The subnetwork allocation of complete WK-Recursive networks has been discussed [4, 13]. On the other hand, various variations of the complete WK-Recursive networks have been proposed. *Three-dimensional WK-Recursive networks* are defined; and a performance comparison of standard WK-Recursive networks and three-dimensional WK-Recursive networks is given [3]. *Hierarchical WK-Recursive networks* and *Pyramid WK-Recursive networks* have been proposed and studied [5]. The *incomplete WK-Recursive networks* have been defined; and the shortest routing algorithm has been devised [9].

It is widely recognized that interprocessor communication is one of the most important issues for interconnection networks because the communication problem is the key issue to many parallel algorithms [8]. *Broadcasting* which is a primitive communication problem is to distribute the same message from a source node to all other nodes without redundancy. The common approach to implement broadcasting algorithm is to embed the broadcasting tree that is a spanning tree with the source node as the root [6]. Numerous applications employ a broadcasting algorithm as a basic function. For example, it is applied in the applications such as matrix operations (e.g., matrix multiplication, factorization, inversion, transposition), database

operations (e.g., polling, master-slave operation), transitive closure algorithms, distributed fault diagnosis, distributed agreement and distributed election [6, 14]. The interconnection network must facilitate efficient broadcasting algorithm to achieve high performance during execution of the various applications. A broadcasting algorithm is *distributed* if it is distributed among all the nodes in the interconnection networks; and each node is responsible for deciding the neighbors to forward the broadcasting message by its own decisions except a centralized controlling node.

Chen and Duh have proposed a distributed stack-base broadcasting algorithm for the complete WK-Recursive networks [1]. Moreover, the algorithm has been generalized for other relative networks [2]. To perform this algorithm, a stack of length L+1, which is used to keep the labels of links, is included in the broadcasting message. It is forbidden to further transmit the message through the links whose labels appear in the stack. Initially, the source node pushes the label L into the stack and transmits the message through all its incident links except the free link. Each node, after it has received the message through a link labeled i, where $1 \le i \le L$-1, performs the following: (1) pop the elements of the stack until the current top element is greater than i; (2) push i into the stack and (3) transmits the message through the links whose labels do not appear in the stack. Clearly, it requires L operations to decide the neighbors to transmit at most, and L is O(log N) where N is the number of nodes. Recall that the length of the stack is also O(L) = O(log N). Thus the distributed stack-base broadcasting algorithm requires O(log N) element in the message and O(log N) time to decide the neighbors to transmit. Because the broadcasting is a primitive problem on the interconnection networks, its performance reveals particular importance. For example, the distributed s broadcasting algorithm on the well-known hypercube requires only constant time to determine the neighbors to forward the message [8]. In this paper, we present a novel broadcasting algorithm for the complete WK-Recursive networks. Our algorithm which is much simpler requires only constant data included in each message and constant time to determine the neighbors to forward the message.

2 Notations and background

A *complete graph* with n nodes, denoted by K_n, is a graph in which every two distinct nodes are adjacent. A WK(W, L), where $W \ge 2$, can be recursively constructed as: WK(W, 0) is a node with W free links that are not incident to other nodes yet. WK(W, 1) is a K_W in which each node has one free link and W-1 links that are used for connecting to other nodes. Clearly, WK(W, 1) has W nodes and W free links. WK(W, C) consists of W copies of WK(W, C-1) as supernodes and the W supernodes are connected as a K_W, where $2 \le C \le L$. By induction, it is easy to see that WK(W, L) has W^L nodes and W free links. Consequently, for any specified number of degree W, the complete WK-Recursive networks can be expanded to arbitrary level L without reconfiguring the links. In Figure 1 and Figure 2, the structures of the WK(4, 0), the WK(4, 1), the WK(4, 2) and the WK(4, 3) are shown.

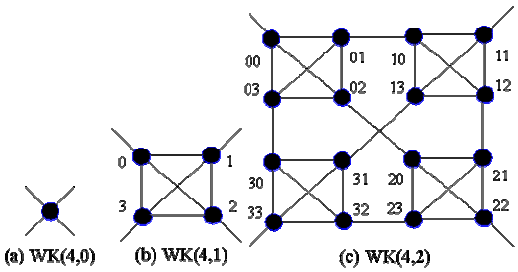

Figure 1: The structures of the WK(4, 0), the WK(4, 1) and the WK(4, 2).

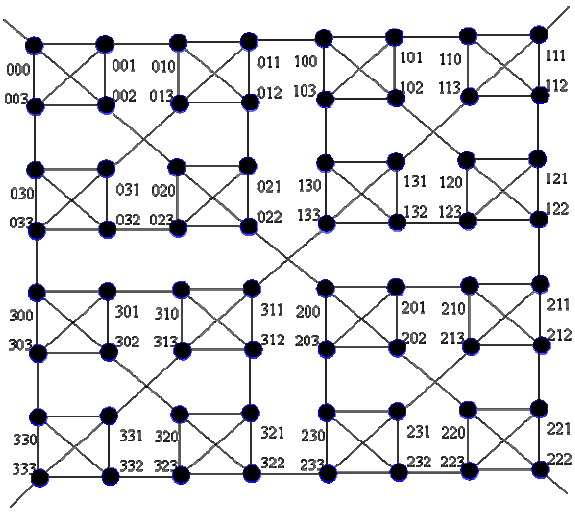

Figure 2: The structures of the WK(4, 3).

In order to utilize the full bandwidth, we assume that each node can transmit and receive messages along different incident links simultaneously (i.e., *all port* assumption). This assumption is quite reasonable because the time required for local computation is negligible. In fact, if each node can transmit and receive messages along only one incident link at any one time (i.e., *one port* assumption), the bandwidth of any interconnection network is the same as the bandwidth of a ring [8].

The rest of this paper is organized as follows. In Section 2, we present some notations and background that will be used throughout this paper. In Section 3, we propose our broadcasting algorithm for the complete WK-Recursive networks. The paper is concluded in Section 4.

The following addressing scheme for WK(W, L) is described in [10]. After fixing an origin and an orientation (i.e., clockwise or counterclockwise), each node within a WK(W, 1) subnetwork is labeled with an index digit d_1 from 0 to W-1. Similarly, each WK(W, C-1) subnetwork within a WK(W, C) subnetwork is

labelled with an index d_C from 0 to W-1, where $2 \le C \le L$. Hence, each node of WK(W, L) is labeled with an unique address $d_L d_{L-1} \ldots d_2 d_1$ as illustrated in Figure 1 and Figure 2. Likewise, a subnetwork of WK(W, L) can be represented by a string $d_L d_{L-1} \ldots d_{C+1}(*)^C$ over set $\{0,1,\ldots,W\text{-}1\} \cup \{*\}$, where * is a "don't care" symbol and $(*)^C$ represents C consecutive *'s. For example, in WK(4, 3), 0** is the subnetwork $\{0 d_2 d_1 \mid 0 \le d_2 \le 3$ and $0 \le d_1 \le 3\}$.

For a subnetwork $d_L d_{L-1} \ldots d_{C+1}(*)^C$ in WK(W, L), a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$ is called a *corner node* of $d_L d_{L-1} \ldots d_{C+1}(*)^C$. For example, in WK(4, 3), 000, 011, 022 and 033 are corner nodes of 0**. Specifically, the node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$ is called the d_C-*corner* of $d_L d_{L-1} \ldots d_{C+1}(*)^C$. For example, in WK(4, 3), the nodes 000, 011, 022 and 033 are 0-corner, 1-corner, 2-corner and 3-corner of 0**, respectively. Note that node 000 is also the 0-corner of 00* and WK(4, 3).

Definition 1. The *corner identifier* of a node $v = d_L d_{L-1} \ldots d_2 d_1$, denoted by *cor-id*($v$), is defined as d_1.

Definition 2. The *corner level* of a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, denoted by *cor-level*(v) is defined as C.

For example, the corner identifiers of nodes 000, 011, 022 and 033 are 0, 1, 2 and 3, respectively. In fact, a node is the *d-corner* of a subnetwork if and only if the corner identifier of the node is d. The corner levels of nodes 000, 011, 012 are 3, 2, and 1, respectively. Note that each node has a corner identifier between 0 and W-1 and a corner level between 1 and L in WK(W, L).

In this paper, a link within a WK(W, 1) subnetwork is called an *inner-cluster link*.

Definition 3. The inner-cluster links of node $d_L d_{L-1} \ldots d_2 d_1$ are defined as $(d_L d_{L-1} \ldots d_2 d_1, d_L d_{L-1} \ldots d_2 h)$, where $0 \le h \le W$-1 and $d_1 \ne h$.

For example, in WK(4, 3), (002, 000), (002, 001), (002, 003) are inner-cluster links of node 002. Clearly, each node has W-1 inner-cluster links in WK(W, L). A link connecting two WK(W, C) subnetworks, where $1 \le C \le L$-1, is called an *inter-cluster link* and specifically a *C-level link*.

Definition 4. The C-level inter-cluster link of node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, is defined as $(d_L d_{L-1} \ldots d_{C+1}(d_C)^C,\ d_L d_{L-1} \ldots d_C (d_{C+1})^C)$. The *inter-cluster neighbor* of a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, is $d_L d_{L-1} \ldots d_C (d_{C+1})^C$. The *flipping corner identifier* of a node $v = d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, denoted by *flip-cor-id*(v), is the corner identifier of inter-cluster neighbor of v.

For example, in WK(4, 3), (022, 200) is a 2-level link and (012, 021) is a 1-level link. Note that each node except the corner nodes $(d_L)^L$, where $0 \le d_L \le W$-1, has exactly one inter-cluster link in WK(W, L). Each corner node $(d_L)^L$ of WK(W, L) has no inter-cluster link but a free link. In fact, a node is incident to a C-level inter-cluster link, where $1 \le C \le L$-1, if and only if the corner level of the node is C. In this paper, a node v is said to be connected to a subnetwork U if there exists a node $u \in U$ such that v is adjacent to u. Nodes 133 and 311 are inter-cluster neighbors of each other. Thus, *flip-cor-id*(133) = 1 and *flip-cor-id*(311) = 3.

Definition 5. The *outline graph* of a WK(W, L), denoted by an OG(WK(W, L)), is to take each WK(W, 1) subnetwork as a supervertex.

Recall that a WK(W, L) can be constructed recursively. If each WK(W, 1) subnetwork of a WK(W, L) is taken as a supervertex, the WK(W, L) will be transformed to a WK(W, L-1). Moreover, each original level-1 inter-cluster link will be an inner-cluster link in the OG(WK(W, L)); and each original level-J inter-cluster link will be a level-(J-1) inter-cluster link in the OG(WK(W, L)), where L-1 $\ge J \ge$ 2. We have the following proposition.

Proposition 1. *An* OG(WK(W, L)) is a WK(W, L-1).

3 The broadcasting algorithm

In this section, we present a new broadcasting algorithm for the complete WK-Recursive networks. Suppose that each node is associated with its own node address, corner identifier, corner level and flipping corner identifier; and each node has kept the system size level L. This algorithm is based on the idea as follows: Let A be a broadcasting algorithm that works for WK(W, L). According to Proposition 1, the outline graph of WK(W, L+1) is WK(W, L). If we apply A to the outline graph of WK(W, L+1), a broadcasting algorithm between these WK(W, 1) subnetworks of WK(W, L+1) is obtained. When a node receives the broadcasting message from an inter-cluster link, it broadcasts the message to the other nodes in the same WK(W, 1) subnetwork. As a consequence, the message can be broadcast to each node of WK(W, L+1) correctly.

3.1 Corner Broadcasting Algorithm

First, we devise an algorithm, *corner broadcasting algorithm*, to deal with the case in which the source node is a corner node $(p)^L$, where $0 \le p \le W$-1, of WK(W, L). Observe that in each subnetwork $c(*)^{L-1}$, where $0 \le c \le W$-1 and $c \ne p$, the node connected to $p(*)^{L-1}$ is its p-corner = $c(p)^{L-1}$. For example, in subnetworks 1*, 2* and 3* of WK(4, 2), the nodes connected to 0* are nodes 10, 20 and 30, respectively; in subnetworks 1**, 2** and 3** of WK(4, 3), the nodes connected to 0** are nodes 100, 200 and 300, respectively. With the aid of this observation, a broadcasting algorithm for source node = $(p)^L$ (i.e., corner broadcasting algorithm) can be developed.

The broadcasting message is included by a label *source-corner-identifier* that records the corner identifier p of the source node. Initially, the source node $(p)^L$ disseminates the message labeled p to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links. Each node when it receives the message labeled p performs by the following conditions:

(1.1) if it receives the message from its inter-cluster link, broadcasts this message to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links.

(1.2) if it *receives* the message from its inner-cluster link,

(1.2.1) if *its* corner level is L, terminates transmitting.

(1.2.2) if its corner level is not L,

(1.2.2.1) if its flipping corner identifier is p, transmits the message to its inter-cluster neighbor by its inter-cluster link.

(1.2.2.2) if its flipping corner identifier is not p, terminates transmitting.

It is clearly that this algorithm works correctly for WK(W, 1) and WK(W, 2). For example, supposed that node 11 is the source node of broadcasting in WK(4, 2). Initially, node 11 broadcasts the message labelled 1 to nodes 10, 12 and 13. Then, according to Condition (1.2.2.1), the nodes 10, 12 and 13 transmit the message labelled 1 to nodes 01, 21 and 31, respectively. According to Condition (1.1), the nodes 01, 21 and 31 broadcast the message labelled 1 to all other nodes in 0*, 2* and 3*, respectively. Because the corner level of the nodes 00, 22 and 33 is 2, according to Condition (1.2.1), they terminate transmitting. Because the corner identifier of the nodes 02, 03, 20, 23, 30 and 32 is not 1, according to Condition (1.2.2.2), they terminate transmitting and duplicate message between 0*, 2* and 3* can be avoided.

Theorem 1. Using corner broadcasting algorithm, starting from a corner node $(p)^L$, a message can be transmitted to each node of WK(W, L) exactly once with 2^L-1 time steps.

Proof. We will prove the lemma by induction on L.

Clearly, it is true for L =1, 2.

Hypothesis: Assume that it is true for $L = k$.

Induction Step: Suppose that the source node is $(p)^{k+1}$. By hypothesis, the message labelled p can be transmitted to each node of $p(*)^k$ exactly once with 2^k-1 time steps. Then, according to Condition (1.2.2.1), each c-corner of $p(*)^k$ (i.e., $p(c)^k$), where $0 \le c \le W$-1 and $c \ne p$, will transmit the message labelled p to their inter-cluster neighbors, $c(p)^k$, respectively; because the flipping corner identifier of $p(c)^k$ is p. By hypothesis, in each $c(*)^k$, starting from the $c(p)^k$, the message labeled p can be transmitted to each node of $c(*)^k$ exactly once with 2^k-1 time steps. Thus, total time steps for this broadcasting is $2(2^k\text{-}1)+1 = 2^{k+1}\text{-}1$. This extends the induction and completes the proof. Q. E. D.

3.2 General Broadcasting Algorithm

Based on the corner broadcasting algorithm, we propose a *general broadcasting algorithm* to deal with general cases of the broadcasting problem for the complete WK-Recursive networks. To express the idea of this algorithm, the node set of WK(W, L) is partitioned into subsets according to where the source node s resides in. We define S_i, where $1 \le i \le L$, to be {v| if v and s reside in the same WK(W, i) subnetwork but distinct WK(W, i-1) subnetworks}. Particularly, we define $S_0 = \{s\}$. It is easy to see that S_i, where $1 \le i \le L$, consists of W-1 copies of WK(W, i-1) subnetworks. For example, suppose that the source node is node 201 in WK(4, 3). S_0 = {201}, S_1 = {200, 202, 203}, S_2 = {21*, 22*, 23*}, S_3 = {0**, 1**, 3**}.

In stage i, where $1 \le i \le L$, the message has been broadcast to S_{i-1} (i.e., the WK(W, i-1) subnetwork where source node s resides in) in previous stage; and it will be broadcast to S_i (i.e., the other W-1 copies of WK(W, i-1) subnetworks of the WK(W, i) subnetwork where source node s resides in) in this stage. The general broadcasting algorithm is based on the idea as follows: in stage i, first, let the message be transmitted to a corner node of each WK(W, i-1) subnetwork. Second, in each WK(W, i-1) subnetwork, starting from the corner node, broadcasting the message to all other nodes in the same subnetwork like applying the corner broadcasting algorithm. For example, suppose that starting from node 201 in WK(4, 3). In stage 1, node 201 broadcasts the message to nodes 200, 202 and 203. In stage 2, nodes 201, 202 and 203 transmit the message to nodes 210, 220 and 230, respectively; and then they broadcast the message to all other nodes in 21*, 22* and 23*, respectively. In stage 3, nodes 200, 211 and 233 transmit the message to nodes 022, 122 and 322, respectively; and then they broadcast the message to all other nodes in 0**, 1** and 3** like applying the corner broadcasting algorithm, respectively.

The broadcasting message is included by a label (m, t), where m records the max level of link that it has ever passed and t records the corner identifier of the WK(W, i-1) subnetwork in each stage i, where $1 \le i \le L$. Clearly, in stage i, the m included in each message is i-1. Initially, the source node s broadcasts the message labelled (0, *cor-id*(s)) (i.e., m = 0 and t = *cor-id*(s)) to the other nodes in the same WK(W, 1) subnetwork (i.e., S_1); and if its inter-cluster link exists, transmits the message labelled (*cor-level*(s), *cor-id*(s)) to its inter-cluster neighbor. Each node v when it receives the message labelled (m, t) performs by the following conditions:

(2.1) If it receives the message from its inter-cluster link, it broadcasts the message labelled (m, t) to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links.

(2.2) If it receives the message from its inner-cluster link,

(2.2.1) if cor-level(v) = L, terminates transmitting.

(2.2.2) if L > *cor*-level(v) > m, transmits the message labelled (*cor-level*(v), *flip-cor-id*(v)) to its inter-cluster neighbor by its inter-cluster link.

(2.2.3) if cor-level(v) = m, terminates transmitting.

(2.2.4) if $m > cor$-level(v),

(2.2.4.1) if flip-cor-id(v) = t, transmits the message labeled (m, t) to its inter-cluster neighbor by its inter-cluster link.

(2.2.4.2) if flip-cor-id(v) ≠ t, terminates transmitting.

Condition (2.1) means that if a node of a WK(W, 1) subnetwork has received the message, all other nodes in the same WK(W, 1) subnetwork will receive the message. Condition (2.2) means that if a node receives the message from its inner-cluster link, only its inter-cluster neighbor is under consideration to transmit the message. Thus, no duplicate message transmitting in a WK(W, 1) subnetwork can be guaranteed. Recall that in stage i, where $1 \le i \le L$, the message has been broadcast to S_{i-1} (i.e., the WK(W, i-1) subnetwork where the source node s resides in) in previous stage. First, let the message be transmitted to the corner nodes of other WK(W, i-1) subnetworks. Clearly, m included in the message is i-1. Condition (2.2.4) means that if the m included in the message is less than i-1, it performs like corner broadcasting algorithm. Since there exists no link level greater than or equal to i-1 in a WK(W, i-1) subnetwork, Condition (2.2.4) guarantees that the message can be transmitted to each node in a WK(W, i-1) subnetwork exactly once. Observe that the link level of links which connect two WK(W, i-1) subnetworks is i-1. Recall that in stage i, the m included in the message is also i-1. Condition (2.2.3) means that it terminates transmitting if the corner level equals to m. Thus, duplicate message transmitted between the WK(W, i-1) subnetworks can be avoided. Condition (2.2.2) means that if the corner level, cor-$level$(v), is greater than m, the message labelled by (cor-$level$(v), $flip$-cor-id(v)) should be transmitted to its inter-cluster neighbor in stage cor-$level$(v)+1. Condition (2.2.1) means that it terminates transmitting when a corner node of WK(W, L) receives the message.

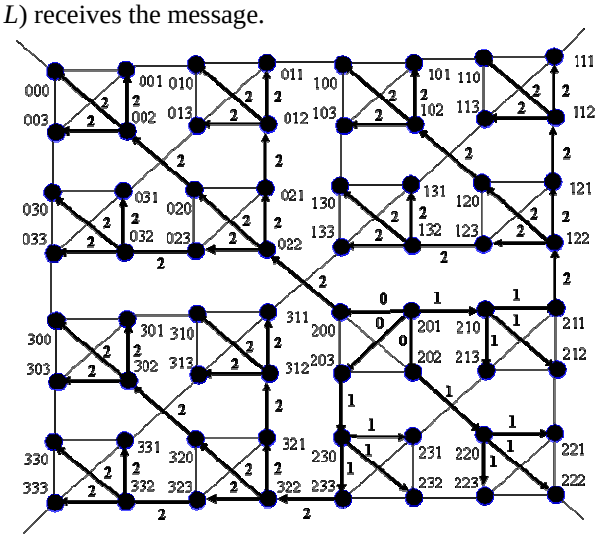

Figure 3: Starting from node 201, broadcasting in WK(4, 3). The label associated with an edge is the m included in the message passed through the edge.

As illustrated in Figure 3, we show an example of broadcasting by applying general broadcasting algorithm. Suppose that node 201 is the source node of broadcasting in WK(4, 3). For the readability of this paper, we describe the broadcasting stage by stage. Initially, in stage 1, node 201 broadcasts the message labelled (0, 1) to nodes 200, 202 and 203. In stage 2, according to Condition (2.2.2), because the corner level of the nodes 201, 202 and 203 is 1, they transmit the message labelled (1, 0) to nodes 210, 220 and 230. Then, according to Condition (2.1), the nodes 210, 220 and 230 broadcast the message labelled (1, 0) to all other nodes of 21*, 22* and 23*, respectively. The corner level of 212, 213, 221, 223, 231, 232 is 1. According to Condition (2.2.3), they terminate transmitting the message. Thus, transmitting duplicate messages between 21*, 22* and 23* is avoided. The corner level of 222 is 3. According to Condition (2.2.1), it terminates transmitting the message. In stage 3, because the corner level of nodes 200, 211 and 233 is 2, according to Condition (2.2.2), they transmit the message labelled (2, 2) to nodes 022, 122 and 322, respectively. In what follows, we only describe the broadcasting starting from 022 in 0**. According to Condition (2.1), the node 022 broadcasts the message labelled (2, 2) to other nodes in 02*. According to Condition (2.2.4.1), because the flip corner identifiers of nodes 020, 021 and 023 are 2, they transmit the message labelled (2, 2) to nodes 002, 012 and 032, respectively. According to Condition (2.1), 002, 012 and 032 transmit the message labelled by (2, 2) to all other nodes in 00*, 01* and 03*, respectively. The corner levels of 001, 003, 010, 013, 030, 031 are 1 and their flipping corner identifiers are not 2. According to Condition (2.2.4.2), they terminate transmitting; and duplicate messages between 00*, 01* and 03* can be avoided. The corner levels of 011, 033 are 2, according to Condition (2.2.3), they terminate transmitting and avoid duplicate messages sent to 1** and 3**. Since corner level of node 000 is 3, according to Condition (2.2.1), it terminates transmitting. Similarly, nodes 122 and 322 broadcast the message to all other nodes of 1** and 3**, respectively.

For readability of this paper, we have described general broadcasting algorithm stage by stage. However, we emphasize that it can also be executed in an asynchronous environment because it can be correctly implemented in each node as local computation and communication.

Theorem 2. By applying general broadcasting algorithm, starting from an arbitrary node, a message can be transmitted to each node of WK(W, L) exactly once with 2^L-1 time steps.

Proof. We will prove the theorem by induction on L.

Clearly, it is true for L =1.

Hypothesis: Assume that it is true for $L = k$.

Induction Step: Suppose that the source node is $s = s_{k+1}s_k \ldots s_2s_1$. By hypothesis, the message can be

transmitted to each node of $s_{k+1}(*)^k$ exactly once with 2^k-1 time steps. According to Condition (2.2.2), each c-corner of $s_{k+1}(*)^k$ (i.e., $s_{k+1}(c)^k$), where $0 \le c \le W$-1 and $c \ne s_{k+1}$, will transmit the message labelled (k, s_{k+1}) to their inter-cluster neighbors, $c(s_{k+1})^k$, respectively. According to Condition (2.2.4), the message can be broadcast in each $c(*)^k$ WK(W, k) subnetwork such that each node receives the message exactly once like applying the corner broadcasting algorithm. Moreover, according to Condition (2.2.3), as the message reaches the corner nodes of these WK(W, k) subnetworks, if the corner node is incident to another WK(W, k) subnetwork, they terminate transmitting and duplicate messages between these WK(W, k) subnetworks can be avoided. By hypothesis, time required for broadcasting in these WK(W, k) subnetworks is also 2^k-1 steps. Thus, total time steps for this broadcasting is $2(2^k\text{-}1)+1 = 2^{k+1}\text{-}1$. This extends the induction and completes the proof. Q. E. D.

4 Conclusions

In this paper, the author presents a novel broadcasting algorithm for the complete WK-Recursive networks. It gains many advantages as follows. This algorithm can guarantee that each node receives the message exactly once within 2^L-1 time steps, which is the diameter of WK(W, L). It is very simple and easy to be implemented. In fact, it requires only extra one integer included in each message and constant time to decide the neighbors to broadcast in an asynchronous environment.

Acknowledgement

This work was supported by the National Science Council of the Republic of China under the contract number: NSC95-2221-E-142-003.

References

[1] G. H. Chen and D. R. Duh (1994), Topological properties, communication, and computation on WK-Recursive networks, *Networks*, 24 303-317.

[2] G. H. Chen, S. C. Hwang, H. L. Huang, M. Y. Su and D. R. Duh (2001), A general broadcasting scheme for recursive networks with complete connection, *Parallel Computing*, 27(9) 1273-1278.

[3] R. Fernandes (1992), Recursive interconnection networks for multicomputer networks, *Proceed. Int. Conf. Parallel Process.*, 1 76-79.

[4] R. *Fernandes* and A. Kanevsky (1993), Substructure allocation in recursive interconnection networks, *Proceed. Int. Conf. Parallel* Process., 1 319-322.

[5] R. *Fernandes* and A. Kanevsky, Hierarchical WK-Recursive topologies for multicomputer systems (1993), *Proceed. Int. Conf. Parallel Process.*, 1 315-318.

[6] S. L. *Johnsson* and C. T. Ho (1989), Optimum broadcasting and personalized communication in hypercubes, *IEEE Transaction on Computers*, 38(9) 1249-1268.

[7] A. I. *Mahdaly*, H. T. Mouftah and N. N. Hanna (1990), Topological properties of WK-Recursive networks, *Proceed. Second* IEEE *Workshop on Future Trends of Distributed Computing Systems*, 374-380.

[8] Y. Saad and M. H. Schultz (1989), Data communication in hypercubes, *Journal of Parallel and Distributed Computing*, 6 115-135.

[9] M. Y. *Su*, G. H. Chen and D. R. Duh (1997), A shortest-path routing algorithm for incomplete WK-Recursive networks, *IEEE* Transactions *on Parallel and Distributed Systems*, 8(4) 367-379.

[10] G. D. Vecchia and C. Sanges (1987), Recursively scalable network for message passing architecture, *Proceed. Int. Conf. Parallel Processing and Applications*, 1 33-40.

[11] G. D. Vecchia and C. Sanges (1988), A recursively scalable network VLSI implementation, *Future Generat. Comput. Syst.* 4(3) 235-243.

[12] L. Verdoscia and R. Vaccaro (1999), An adaptive routing algorithm for WK-Recursive topologies, *Computing*, 63(2) 171-184.

[13] F. Wu and C. C. Hsu (2002), A generalized processor allocation scheme for recursively decomposable interconnection networks, IEICE *Transaction on Information and Systems*, E85D(4) 694-713.

[14] J. Wu and E. B. Fernandez (1993), Fault-tolerant distributed broadcast algorithm for cube-connected-cycles, *Computer Systems Science and Engineering*, 4 224-233.