Journal of Graph Algorithms and Applications
http://www.cs.brown.edu/publications/jgaa/
vol. 2, no. 2, pp. 1–17 (1998)

A Broadcasting Algorithm with Time and Message Optimum on Arrangement Graphs

Leqiang Bai   *Hajime Maeda*
Department of Communication, Osaka University
http://nanase.comm.eng.osaka-u.ac.jp/personal/maeda.html/
leqiang@comm.eng.osaka-u.ac.jp   h-maeda@comm.eng.osaka-u.ac.jp

Hiroyuki Ebara
Faculty of Engineering, Kansai University
ebara@eip.kansai-u.ac.jp

Hideo Nakano
Media Center, Osaka City University
http://www.media.osaka-cu.ac.jp/ nakano/
nakano@media.osaka-cu.ac.jp

Abstract

In this paper, we propose a distributed broadcasting algorithm with optimal time complexity and without message redundancy for one-to-all broadcasting in the one-port communication model on arrangement graph interconnection networks. The algorithm exploits the hierarchical property of the arrangement graph to construct different-sized broadcasting trees for different-sized subgraphs. These different-sized broadcasting trees constitute a spanning tree on the arrangement graph. Every processor individually performs its broadcasting procedure based on the spanning tree. It is shown that a message can be broadcast to all the other $\frac{n!}{(n-k)!} - 1$ processors in at most $O(k \lg n)$ steps on the (n, k)-arrangement graph interconnection network. The algorithm can also guarantee that each of processors on the arrangement graph interconnection network receives the message exactly once.

Communicated by Alon Itai: submitted June 1996; revised May 1998.

1 Introduction

The star graph [1] is one of the widely studied interconnection network topologies. It has been proposed as an attractive alternative to the hypercube with many superior characteristics. A major practical difficulty with the star graph is related to its number of nodes: $n!$ for the $n-$star graph. Recently, a new interconnection network topology called the (n, k)-arrangement graph has been proposed in [5]. This topology is a generalized class of star graphs in the sense that a star graph is a special arrangement graph, and presents a cure for the design drawback of the star graph. It brings a solution to the problem of growth of the number $n!$ of nodes in the $n-$star graph with respect to the dimension n. Namely, the (n, k)-arrangement graph has more flexibility in selecting the design parameters: size, diameter, and degree than the star graph. It also keeps all the desirable topological qualities of the star graph topology such as hierarchical structure, vertex and edge symmetry, simple routing and many fault tolerance properties.

Broadcasting is one of the fundamental communication problems for distributed memory interconnection networks. In broadcasting, one processor (or node) has a message which needs to be communicated to everyone; such a processor is called the source of broadcasting and every other node to which the message needs to be sent is called the destination of broadcasting. Broadcasting is a very important operation used in various linear algebra algorithms, database queries, transitive closure algorithms, and linear programming algorithms. The interconnection network must facilitate efficient broadcasting so as to achieve high performance during execution of jobs.

The efficiency of the broadcasting algorithms is characterized by the time complexity, the number of steps required, and the message complexity, the total number of messages exchanged, to complete the broadcasting. Hence, it is desirable to develop a broadcasting algorithm that optimizes both the time complexity and the message complexity. The broadcasting problems on the hypercube and the star graph have been investigated in recent years. In [8], Johnsson and Ho presented three new communication graphs for hypercubes and defined scheduling disciplines, so that the communication tasks are completed within a small constant factor of the best known lower bounds. In [11], Sheu, Liaw and Chen presented a distributed broadcasting algorithm without message redundantly in star graphs. It takes $2n - 3$ steps in the multi-port communication model. In [9], Mendia and Sarkar proposed a broadcasting algorithm with the optimal time complexity in the one-port communication model on the star graph. It exploits the rich structure of the star graph and works by recursively partitioning the original star graph into smaller star graphs. In [12], Sheu, Wu and Chen proposed a broadcasting algorithm that broadcasts a message to all nodes in the star graph at the optimal time based on the algorithm in [9]. It also performs broadcasting without redundant messages. In [4], Bai, Yamakawa, Ebara and Nakano proposed a broadcasting algorithm in the one-port communication model on the arrangement graph. It can broadcast a message to all nodes in optimal time. But the message complexity of the algorithm is not optimal.

In this paper, we consider the one-to-all broadcasting problem in the one-port communication model on the arrangement graph and propose a distributed broadcasting algorithm in the message passing mode. By exploiting the rich topological properties of the arrangement graph to construct different sized broadcasting trees for different sized subgraphs recursively, this algorithm can broadcast a message to all the other nodes on the arrangement graph in the optimal time complexity in the sense of $O-$notation and without message redundancy. The remainder of this paper is organized as follows. The arrangement graph , its basic properties and some definitions are introduced in Section 2. The broadcasting algorithm is described in Section 3. The conclusion is given in Section 4.

2 Preliminaries

Let n and k with $1 \leq k \leq n$ be two integers, and let us write $< n >= \{1, 2, \ldots, n\}$ and $< k >= \{1, 2, \ldots, k\}$. Let P_n^k be the set of permutations of k elements chosen from $< n >$. The k elements of a permutation p are denoted $p_1, p_2, \ldots, p_k$; we write $p = p_1 p_2 \ldots p_k$.

Definition 1 *The (n, k)-arrangement graph $A_{n,k} = (V, E)$ is an undirected graph given by:*

$$\begin{aligned} V &= \{p_1 p_2 \ldots p_k \,|\, p_i \, in \, < n > \, and \, p_i \neq p_{i'} \, for \, i \neq i'\} = P_n^k, \quad and \\ E &= \{(p, q) |\, p \, and \, q \, in \, V \, and \, for \, some \, i \in < k >, \, p_i \neq q_i \\ &\qquad\qquad\qquad\qquad and \, p_{i'} = q_{i'} \, for \, i \neq i'\} \ . \end{aligned} \tag{1}$$

The $(4, 2)-$arrangement graph is shown in Figure 1. It illustrates that $A_{4,2}$ can be decomposed into 4 smaller subgraphs, where each of them has the fixed element i for $1 \leq i \leq 4$ in position 2. Each of subgraphs contains three nodes.

The $(n, k)-$arrangement graph is a regular graph of degree $k(n - k)$, the number of nodes $\frac{n!}{(n-k)!}$, and diameter $\lfloor \frac{3}{2} k \rfloor$. Notice that there are $\frac{(n-1)!}{(n-k)!}$ nodes in $A_{n,k}$ which have the element p_i in position i for any fixed $p_i \in < n >$ and $i \in < k >$. These nodes form a subgraph of $A_{n,k}$. The arrangement graph still has many other good characteristics. For a more thorough coverage of the arrangement graph, refer to [5].

As shown in Definition 1, there are $n - k$ elements that are not used in permutation p. We define the label p_{out} of the node p to denote these elements. Using the label p_{out}, we define the operator g_{ij} that carries out the permutation p and the label p_{out} of the node p. In this way, we represent the node p and its edges on the arrangement graph so that the node p corresponds to the

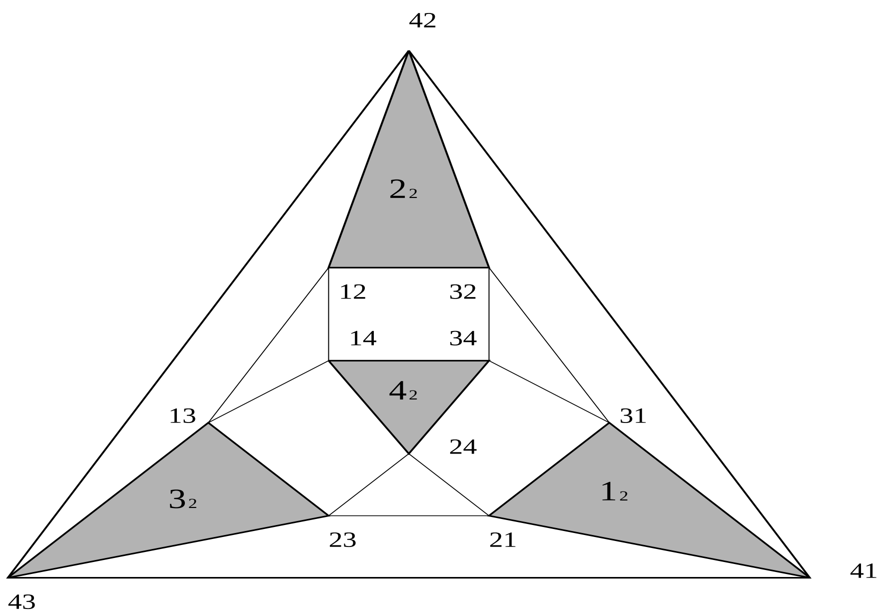

Figure 1: (4,2)- arrangement graph

permutation p with a label p_{out}, and its edges correspond to the actions of the operators on the permutation p and the label p_{out}.

Let $INT(p) = \{p_1, p_2, \ldots, p_k\}$, which is the set of k elements of $< n >$ used in permutation p, and let $EXT(p) =< n > -INT(p)$, which is the set of $n - k$ elements of $< n >$ not used in permutation p.

Definition 2 *For each node p, define P_{out}, which we call the set of labels p_{out} of p, to be the set of elements of P_n^{n-k} given:*

$$P_{out} = \{p_{k+1} \ldots p_j \ldots p_n | \, p_j \, in \, EXT(p) \, for \, k+1 \leq j \leq n \\ and \, p_{j_1} \neq p_{j_2} \, for \, j_1 \neq j_2\} \quad . \tag{2}$$

Example. Consider $< n >= \{1, 2, 3, 4, 5\}$, $< k >= \{1, 2, 3\}$ and $p = 245$. Then, $P_{out} = \{13, 31\}$ and the two possible values of p_{out} are 13 and 31.

Definition 3 *The operator $g_{ij}(p, p_{out})$: $P_n^k \rightarrow P_n^k$ takes a node p to its adjacent node p′ where p′ and p differ only in the ith position, and the element in the ith position of p′ is the element in the jth position of p_{out}, $1 \leq i \leq k$, $k+1 \leq j \leq n$. We define g_0 to be the identity operator, and $pg_0 = p$ for all p and p_{out}.*

Example. Let $p = 123$ be a node in $A_{5,3}$ and be acted on by g_{15}. For the different $p_{out} = 45$ or 54, $pg_{15} = 523$ or 423.

From the definitions of operator and label, we know that the operator g_{ij} of the node p has the different actions for the different p_{out} of the node p. In this paper, we assume that the source node has a given p_{out} and any node acted on by the operator acts on its p_{out}; any node revises its p_{out} based on the received message and uses this p_{out} before receiving a new message.

Let $First_i(p)$ have as its value the first i elements of the node p, that is, $First_i(p) = p_1p_2...p_i$, $1 \leq i \leq k$, and $First_0(p) = \lambda$, where λ is the empty string. For given $p = p_1p_2......p_k$ and $p_{out} = p_{k+1}...p_n$, let $Elect_i(p, p_{out})$ have as its value the ith element of p and p_{out}, that is, $Elect_i(p, p_{out}) = p_i$, $1 \leq i \leq n$. We will write $Elect_i(p)$ when p_{out} is understood. Here, we give two definitions of subgraphs based on permutation p and its label p_{out} on the arrangement graph. We say $A^l_{n,k}$ the lth level subgraph of $A_{n,k}$ and the level $l+1$ is lower than the level l.

Definition 4 *The induced subgraph $A^l_{n,k}(First_l(p))$, $1 \leq l \leq k$, of $A_{n,k}$ for a node p, is the induced subgraph having as vertices the set of the nodes whose the first l elements are $p_1p_2...p_l$, and $A^0_{n,k} = A_{n,k}$. The set $V^l_{n,k}(First_l(p))$ of the nodes in $A^l_{n,k}(First_l(p))$ is given by:*

$$\begin{aligned} V^l_{n,k}(First_l(p)) = \{ & p_1 \dots p_l \, p_{l+1} \dots p_i \dots p_k | \\ & p_i \in < n > - \{p_1, ..., p_l\} \ , \ l+1 \leq i \leq k\} \in V \ . \end{aligned} \tag{3}$$

Example. $V^1_{5,3}(2) = \{2pq | p, q \neq 2\}$ is the set of the nodes of the subgraph $A^1_{5,3}(2)$ on $A_{5,3}$, where each of the nodes in the subgraph $A^1_{5,3}(2)$ has the element 2 in the first position.

Definition 5 *We will write, for p and its p_{out}, the expression $A^l_{n,k}(Elect_i(p))$ for $A^l_{n,k}(First_l(q))$, where $q = p_1...p_{l-1}p_i$, $l - 1 \leq i \leq n$.*

Example. Let $p = 123$, with its $p_{out} = 45$, be a node in $A_{5,3}$. Then $First_1(p) = 1$ and $Elect_4(p) = 4$. Each of the nodes in $A^1_{5,3}(First_1(p)) = A^1_{5,3}(1)$ has the element 1 in position 1. Each of the nodes in $A^2_{5,3}(Elect_4(p)) = A^2_{5,3}(First_1(p)4) = A^2_{5,3}(14)$ has the element 4 in position 2 and the element 1 in position 1. $V_{5,3} = \{V^1_{5,3}(1), V^1_{5,3}(2), V^1_{5,3}(3), V^1_{5,3}(4), V^1_{5,3}(5)\}$.

Definitions 4 and 5 mean that $A^l_{n,k}(First_l(p))$ and $A^l_{n,k}(Elect_i(p))$ are defined based on the node p, p_{out} pair. Each of the nodes in $A^l_{n,k}(First_l(p))$ has the same l elements in the first l positions as the node p has. Each of the nodes in $A^l_{n,k}(Elect_i(p))$ has the element $Elect_i(p)$ in position l and the same $l - 1$ elements in the first $l - 1$ positions as the node p has.

3 Broadcasting Algorithm

In this section, we consider one-to-all broadcasting problem on a fault-free arrangement graph $A_{n,k}$, and develop a broadcasting algorithm with the optimal time complexity in the sense of $O-$notation and without message redundancy. We assume that a node consists of a processor with bidirectional communication links to each of its adjacent nodes. Any node knows the condition of its adjacent links and has a large enough buffer to preserve the message that it sends and received. At any given time, a node can communicate with at most one of its adjacent nodes. There are no faults on the arrangement graph.

Theorem 1 [10] *If an interconnection network consists of N nodes or processors that can communicate with at most one of its adjacent nodes at any given time, then any one-to-all broadcasting algorithm on the network must take at least $\Omega(\lg N)$ steps.*

The (n, k)-arrangement graph is a graph with hierarchical structure. Since we use recursively its hierarchical property to develop our broadcasting algorithm on $A_{n,k}$, we consider the broadcasting procedure on some subgraph $A^l_{n,k}$ of $A_{n,k}$, $0 \le l \le k$. Let p be a source node in some $A^l_{n,k}$ and be ready to broadcast a message to its $n-l$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$, $l+1 \le i \le n$, of $A^l_{n,k}(First_l(p))$. According to the relationship of p and $n-l$ subgraphs $A^{l+1}_{n,k}$ of $A^l_{n,k}(First_l(p))$, $n-l$ subgraphs $A^{l+1}_{n,k}$ can be divided into the following three cases:

case 1: $A^{l+1}_{n,k}(Elect_{(l+1)}(p))$ in which p is.

case 2: $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, which are not directly connected to p.

case 3: $A^{l+1}_{n,k}(Elect_i(p))$, $k+1 \le i \le n$, which are directly connected to p.

Since the node p is in $A^{l+1}_{n,k}(Elect_{l+1}(p))$, we only need to send the message to $k-l-1$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, and $n-k$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$, $k+1 \le i \le n$. In case 2, p is not directly connected to any node that belongs to $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$. This is because the element $Elect_i(p)$ in position $l+1$ of the nodes in $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, is in $INT(p)$. To send a message to some subgraphs of $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, we have to send the message to some intermediate nodes that are directly connected to some nodes in $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, and then send the message to these subgraphs. In case 3, some nodes in $A^{l+1}_{n,k}(Elect_i(p))$, $k+1 \le i \le n$, are directly connected to p since the elements $Elect_i(p)$ in position $l+1$ of the nodes in $A^{l+1}_{n,k}(Elect_i(p)|k+1 \le i \le n)$ are not in $INT(p)$. If p sends a message to all $n-l$ node-disjoint subgraphs $A^{l+1}_{n,k}$ of $A^{l}_{n,k}(First_l(p))$ sequentially, it takes $O(n-l)$ steps for p to complete this broadcasting procedure. Applying recursively this procedure, it takes $O\{(k-l)(n-l)\}$ steps for p to broadcast a message to all the other nodes in $A^{l}_{n,k}(First_l(p))$. Since there are $N = \frac{(n-l)!}{(n-k)!}$ nodes in $A^{l}_{n,k}(First_l(p))$ and an optimal broadcasting procedure requires at most $O(\lg N) = O((k-l)\lg(n-l))$ steps based on Theorem 1, this broadcasting procedure is not optimal in time complexity.

If we only consider broadcasting the message to at least one of the nodes in each of $n-l$ node-disjoint subgraphs $A^{l+1}_{n,k}$ of $A^{l}_{n,k}(First_l(p))$ and then making each of the nodes that received broadcast the message on the given lower level subgraph, a broadcasting algorithm with optimal time has been proposed in [4]. This algorithm embeds the intermediate nodes in a broadcasting tree to achieve a broadcasting procedure with the optimal time complexity. But it is not optimal in the message complexity. The reason is that it did not consider that different intermediate nodes belong to different level subgraphs and should broadcast the message on different level subgraphs.

To perform the broadcasting procedure with time and message optimality on $A^{l}_{n,k}(First_l(p))$, we define the intermediate node $p(i)$ that is directly connected to $A^{l+1}_{n,k}(Elect_i(p))$ for $l+2 \le i \le k$ as follows.

Definition 6 *There may be many intermediate nodes. An intermediate node $p(i)$ is a node that satisfies $First_{i-1}(p(i)) = First_{i-1}(p)$ and $Elect_n(p(i)) = Elect_i(p)$ for $l+2 \le i \le k$.*

Lemma 1 *The intermediate node $p(i)$ is adjacent to $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$.*

Proof: For $l+2 \le i \le k$, $p(i)g_{(l+1)n}$ is in $A^{l+1}_{n,k}(Elect_i(p))$. □

Table 1: The broadcasting procedure from p to each of $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le n$.

Phase	the receiving nodes	comments
beginning	p	the node p is in $A^{l+1}_{n,k}(Elect_{l+1}(p))$
1	$p(k)$,..., $p(l+2)$	$p(i)$, $l+2 \le i \le k$, is adjacent to $A^{l+1}_{n,k}(Elect_i(p)))$
	$p(i)g_{(l+1)n}$, $l+2 \le i \le k$,	$p(i)g_{(l+1)n}$, $l+2 \le i \le k$, is in $A^{l+1}_{n,k}(Elect_i(p)))$
2	$p(j)$ $k+1 \le j \le n$	$p(j)$, $k+1 \le j \le n$, is in $A^{l+1}_{n,k}(Elect_j(p))$

As shown in Table 1, we divide the broadcasting procedure for p to broadcast the message to each of $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le n$, into two phases:

Phase 1: The node p and the intermediate nodes $p(i')$, $l+2 \leq i \leq i' \leq k$, broadcast the message to the intermediate nodes $p(i)$, $l+2 \leq i \leq k$. Then, the intermediate nodes $p(i)$, $l+2 \leq i \leq k$, broadcast the message to $A^{l+1}_{n,k}(Elect_i(p))$ at one step.

Phase 2: The node p, the intermediate nodes $p(i)$, $l+2 \leq i \leq k$, and the nodes $p(j')$ in $A^{l+1}_{n,k}(Elect_{j'}(p))$, $k+1 \leq j' \leq j \leq n-1$, broadcast the message to the nodes $p(j)$ in $A^{l+1}_{n,k}(Elect_j(p))$, $k+1 \leq j \leq n-1$. Then, the node p sends the message to $A^{l+1}_{n,k}(Elect_n(p))$ at the last step.

Phase 1. At the beginning of broadcasting procedure, p sends the message to $p(k) = pg_{kn}$. Then, $First_{k-1}(p(k)) = First_{k-1}(p)$ and $Elect_n(p(k)) = k$. At the next step, p and $p(k)$ send the message to $p(k-1) = pg_{(k-1)n}$ and $p(k-2) = p(k)g_{(k-2)n}$ respectively. Then, $First_{k-2}(p(k-1)) = First_{k-2}(p)$ and $First_{k-3}(p(k-2)) = First_{k-3}(p)$. In this way, the message can be sent to $k-l-1$ intermediate nodes $p(i)$, $l+2 \leq i \leq k$. Because each step doubles the number of the nodes that receive the message, there are 2^m nodes that receive the message at the mth step. It takes $O(\lg(k-l))$ steps for p to complete the broadcasting procedure to $k-l-1$ intermediate nodes respectively adjacent to $k-l-1$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$ for $l+2 \leq i \leq k$. Then, the $k-l-1$ intermediate nodes $p(i)$, $l+2 \leq i \leq k$, can send the message to $k-l-1$ subgraphs $A^{l+1}_{n,k}(Elect_i(p))$ respectively at one step.

Phase 2. Because $Elect_j(p(i)) = Elect_j(p)$ for $l+2 \leq i \leq k$ and $k+1 \leq j \leq n-1$, the intermediate node $p(i)$ for $l+2 \leq i \leq k$ is also directly connected to $A^{l+1}_{n,k}(Elect_j(p))$ for $k+1 \leq j \leq n-1$. After $k-l-1$ intermediate nodes $p(i)$, $l+2 \leq i \leq k$, receive the message, we can make the source node and these intermediate nodes send respectively the message to $k-l$ nodes $p(j)$ in $k-l$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_j(p))$, $k+1 \leq j \leq k+(k-l) \leq n-1$. Here, $p(k+1) = pg_{(l+1)(k+1)}$ and $p(k+(i-l)) = p(i)g_{(l+1)(k+(i-l))}$ for $l+2 \leq i \leq k$. Since $Elect_i(p(j)) = Elect_i(p)$ for $k+1 \leq j \leq k+(k-l) < i \leq n-1$, the node $p(j)$ can also send the message to $p(j+2(k-l))$ at the next step if $j+2(k-l) \leq n-1$. Similarly, additional step doubles the number of subgraphs that received the message, it takes $O(\lg \frac{n-k}{k-l})$ steps to complete the broadcasting procedure to $n-k-1$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_j(p))$ for $k+1 \leq j \leq n-1$. Then, p can send the message to $A^{l+1}_{n,k}(Elect_n(p))$ at the last step.

After completing the broadcasting procedures of **Phase 1** and **2**, there is only one node that received the message in each of $A^{l+1}_{n,k}(Elect_i(p))$ for $l+2 \leq i \leq n$. The node in each of $A^{l+1}_{n,k}(Elect_i(p))$ for $l+2 \leq i \leq n$ can be considered a source node to broadcast the message within the given $(l+1)$th level subgraphs. In $A^{l+1}_{n,k}(Elect_{l+1}(p))$, there are $k-l$ nodes that hold the message, which are p and $p(i)$, $l+2 \leq i \leq k$. If they are all considered the sources of $A^{l+1}_{n,k}(Elect_{l+1}(p))$, some nodes in this subgraph will receive the message more than once. To ensure that each node in the broadcasting procedure receives once only, we develop the broadcasting procedure **Phase 3** that can perform the broadcasting in $A_{n,k}^{l+1}(Elect_{l+1}(p))$ by decomposing $A_{n,k}^{l+1}(Elect_{l+1}(p))$ into the different level node-disjoint subgraphs based on p and $p(i)$ for $l+2 \le i \le k$. Here, we first give two Lemmas concerning the decomposing of $A_{n,k}^{l+1}(First_{l+1}(p))$ into the different level subgraphs based on p and $p(i)$ for $l+2 \le i \le k$.

Lemma 2 *Based on p and the intermediate node $p(l+2)$ for $l+2 \le k$, $A_{n,k}^{l+1}(First_{l+1}(p))$ can be decomposed into the $(l+2)$th level node-disjoint subgraphs as follows.*

$$V_{n,k}^{l+1}(First_{l+1}(p)) = \{ \bigcup_{l+2 \le i \le n-1} V_{n,k}^{l+2}(Elect_i(p(l+2)))\} \cup V_{n,k}^{l+2}(First_{l+2}(p)) \ . \tag{4}$$

Proof: Based on Definition 5, $V_{n,k}^{l+2}(Elect_{l+2}(p)) = V_{n,k}^{l+2}(First_{l+2}(p))$. Since $First_{l+1}(p) = First_{l+1}(p(l+2))$ and $Elect_n(p(l+2)) = Elect_{l+2}(p)$ from Definition 6, $\bigcup_{l+3 \le i \le n} Elect_j(p) = \bigcup_{l+2 \le i \le n-1} Elect_i(p(l+2))$. Therefore,

$$\begin{aligned} V_{n,k}^{l+1}(First_{l+1}(p)) &= V_{n,k}^{l+2}(Elect_{l+2}(p)) \cup \{ \bigcup_{l+3 \le i \le n} V_{n,k}^{l+2}(Elect_i(p))\} \\ &= \{ \bigcup_{l+2 \le i \le n-1} V_{n,k}^{l+2}(Elect_i(p(l+2)))\} \cup V_{n,k}^{l+2}(First_{l+2}(p)) \ . \end{aligned}$$

□

From Lemma 2, $p(l+2)$ for $l+2 \le k$ is in $A_{n,k}^{l+2}(Elect_{l+2}(p(l+2)))$ that is one of the subgraphs $A_{n,k}^{l+2}(Elect_i(p))$, $l+2 \le i \le n$, of $A_{n,k}^{l+1}(First_{l+1}(p))$. The node $p(l+2)$ can be considered a source node to broadcast the message on $A_{n,k}^{l+1}(First_{l+1}(p))$. Since $First_{l+2}(p) = First_{l+2}(p(l+3))$ for $l+3 \le n$, $p(l+3)$ is in $A_{n,k}^{l+2}(Elect_n(p(l+3))) = A_{n,k}^{l+2}(First_{l+2}(p))$. Therefore, $p(l+2)$ can apply the broadcasting procedures of **Phase 1** and **2** to broadcast the message to the subgraphs $A_{n,k}^{l+2}(Elect_i(p))$, $l+3 \le i \le n-1$, and remains $A_{n,k}^{l+2}(Elect_n(p(l+2)))$ for $p(l+3)$. To avoid message redundancy, we will recursively use $p(i)$, $l+2 \le i \le k$, to perform broadcasting in $A_{n,k}^{i-1}(Elect_n(p(i))) = A_{n,k}^{i-1}(First_{i-1}(p))$.

Lemma 3 *Based on p and $p(i)$, $l+2 \le i \le k$, $A_{n,k}^{l+1}(First_{l+1}(p))$ can be decomposed into the different level node-disjoint subgraphs as follows:*

$$V_{n,k}^{l+1}(First_{l+1}(p)) = \{ \bigcup_{l+2 \le i \le k, i \le j \le n-1} V_{n,k}^{i}(Elect_j(p(i)))\} \cup p \ . \tag{5}$$

Proof: Recursively applying Lemma 2 until p. □

Table 2 shows the broadcasting procedures of **Phase 3** based on Lemma 3 for $p(i)$, $l+2 \le i \le k$, to broadcast the message to the given different level node-disjoint subgraphs on $A_{n,k}^{l+1}(First_{l+1}(p))$. The node $p(l+2)$ can apply the broadcasting procedures of **Phase 1** and **2** to broadcast the message on $A^{l+1}_{n,k}(First_{l+1}(p))$ and only remains $A^{l+2}_{n,k}(Elect_n(p(l+2)))$ for $p(l+3)$ if $p(l+3) \leq k$, where $A^{l+2}_{n,k}(Elect_n(p(l+2))) = A^{l+2}_{n,k}(First_{l+2}(p))$. Similarly, $p(l+3)$ can broadcast the message on $A^{l+2}_{n,k}(First_{l+2}(p))$ and only remains $A^{l+3}_{n,k}(Elect_n(p(l+3)))$ for $p(l+4)$ if $l+4 \leq k$. Recursively applying the broadcasting procedure that $p(i)$ broadcasts the message on $A^{i-1}_{n,k}(First_{i-1}(p))$ and only remains $A^{i}_{n,k}(Elect_n(p(i)))$ for $p(i+1)$ until p, the message can be broadcast with message redundancy in the given different level node-subgraphs.

Table 2: The broadcasting procedures of the intermediate nodes $p(i)$, $l+2 \leq i \leq k$.

sending node	sending to subgraphs	not sending to subgraph
$p(l+2)$ in $A^{l+1}_{n,k}(First_{l+1}(p))$	$A^{l+2}_{n,k}(Elect_j(p(l+2)))$ $l+2 \leq j \leq n-1$	$A^{l+2}_{n,k}(Elect_n(p(l+2)))$ $= A^{l+2}_{n,k}(First_{l+2}(p))$
$\vdots$	$\vdots$	$\vdots$
$p(i-1)$ in $A^{i-2}_{n,k}(First_{i-2}(p))$	$A^{i-1}_{n,k}(Elect_j(p(i-1)))$ $i-1 \leq j \leq n-1$	$A^{i-1}_{n,k}(Elect_n(p(i-1)))$ $= A^{i-1}_{n,k}(First_{i-1}(p))$
$p(i)$ in $A^{i-1}_{n,k}(First_{i-1}(p))$	$A^{i}_{n,k}(Elect_j(p(i)))$ $i \leq j \leq n-1$	$A^{i}_{n,k}(Elect_n(p(i)))$ $= A^{i}_{n,k}(First_{i}(p))$
$p(i+1)$ in $A^{i}_{n,k}(First_{i}(p))$	$A^{i+1}_{n,k}(Elect_j(p(i+1)))$ $i+1 \leq j \leq n-1$	$A^{i+1}_{n,k}(Elect_n(p(i+1)))$ $= A^{i+1}_{n,k}(First_{i+1}(p))$
$\vdots$	$\vdots$	$\vdots$
$p(k)$ in $A^{k-1}_{n,k}(First_{k-1}(p))$	$A^{k}_{n,k}(Elect_j(p(k)))$ $k \leq j \leq n-1$	$A^{k}_{n,k}(Elect_n(p(k)))$ $= A^{k}_{n,k}(First_{k}(p)) = p$
p no sending		

Now, we will present a broadcasting algorithm with optimal time complexity and without message redundancy for one-to-all broadcasting on an arrangement graph $A^{l}_{n,k}$. Let *Broadcasting_algorithm* denote our one-to-all broadcasting algorithm. We will describe it based on three phases as shown in Table 1 and 2. Since the node $p(j)$, $k+1 \leq j \leq n-1$, is the source node in $A^{l+1}_{n,k}(Elect_j(p(j)))$, we only need to improve **Phase 2** so that the node $p(j)$, $k+1 \leq j \leq n-1$, will start to broadcast the message on $A^{l+1}_{n,k}(Elect_i(p(j)))$ in **Phase 2** after completing the broadcasting procedure from p to each of $A^{l+1}_{n,k}(Elect_i(p))$, $k+1 \leq i \leq n-1$. In the broadcasting procedure, the nodes that hold the message send the broadcasting request to the other nodes. Every node individually starts to perform its broadcasting algorithm while receiving the request. Let M denote a broadcasting request. A request M consists of following six parameters as shown in Table 3 and $M = \{Data, p_{out}, l, i, m, ID\}$. At the beginning of broadcasting, $M = \{Data, p_{out}, l, 0, 0, 0\}$. The formal description of our broadcasting algorithm is given below.

Table 3: Parameters that a request M contains

1	$Data$	Message to be broadcast.
2	p_{out}	The given label to be sent to the receiving node.
3	l	Dimension of subgraph.
4	i	Integer used for determining edges.
5	m	Used for determining the order of broadcasting.
6	ID	Used for determining the phase of node in broadcasting.
	$ID = 0$	The source node.
	$ID = 1$	The message is sent to $p(i)$, $l + 2 \leq i \leq k$.
	$ID = 2$	The message is sent to $p(j)$, $k + 1 \leq j \leq n - 1$.

Broadcasting_algorithm(M)
  var i, n, l, k, m, ID : *integer*; g_{ij} : *operator*;
begin /* p decides its p_{out} based on M and sends M to the receiving node */
  the node p gets M from buffer;
/* begin Phase 1: broadcast M to $p(i)$, $l + 2 \leq i \leq k$ */
  if $ID = 0$ **and** $l \leq k - 2$ **then begin**
    repeat /* the source node broadcasts M to $p(i)$, $l + 2 \leq i \leq k$ */
      $m := m + 1$; $i := k + 1 - 2^{m-1}$;
      if $i \geq l + 2$ **then** *p sends $M(Data, p_{out}, l, i, m, 1)$ to pg_{in}*
    until $i < l + 2$
  end;
  if $ID = 1$ **then begin**
    $I = i$;
    repeat /* $p(i')$, $l + 2 \leq i < i' \leq k$ broadcast M to $p(i)$ */
      $m := m + 1$; $i := I - 2^{m-1}$;
      if $i \geq l + 2$ **then** *p sends $M(Data, p_{out}, l, i, m, 1)$ to pg_{in}*
    until $i < l + 2$;
    p sends $M(Data, p_{out}, l + 1, 0, 0, 0)$ to $pg_{(l+1)n}$
    /* $p(i)$, $l + 2 \leq i \leq k$, broadcast M to $A^{l+1}_{n,k}(Elect_i(p)))$ */
  end;
/* end Phase 1 */
/* begin Phase 2: broadcast M to $p(i)$, $k + 1 \leq i \leq n$,
    and in $A^{l+1}_{n,k}(Elect_i(p))$, $k + 1 \leq i \leq n - 1$ */
  if $ID = 0$ **and** $l \leq k - 1$ **then begin**
    $m := 0$;
    repeat /* the source node broadcasts M to $p(i)$, $k + 1 \leq i \leq n - 1$ */
      $m := m + 1$; $i := k + 1 + (2^{m-1} - 1)(k - l)$;
      if $i \leq n - 1$ **then** *p send $M(Data, p_{out}, l, i, m, 2)$ to $pg_{(l+1)i}$*
    until $i > n - 1$
    p send $M(Data, p_{out}, l + 1, 0, 0, 0)$ to $A^{l+1}_{n,k}(Elect_n(p)))$
    /* sends M to $p(n)$ in $A^{l+1}_{n,k}(Elect_n(p)))$ */
  end;

```
    if ID = 1 then begin
       m := 0;
       repeat /* p(i'), l+2 ≤ i' ≤ k, broadcast M to p(i), k+2 ≤ i ≤ n−1 */
          m := m+1; i := k+I−l+(2^(m−1) −1)(k−l);
          if i ≤ n−1 then p send M(Data, p_out, l, i, m, 2) to pg_(l+1)i
       until i > n−1
    end;
    if ID = 2 then begin
       I := i;
       repeat /* p(i'), k+1 ≤ i' < i ≤ n−1, broadcast M to p(i) */
          m := m+1; i := I+(2^(m−1))(k−l);
          if i ≤ n−1 then p send M(Data, p_out, l, i, m, 2) to pg_(l+1)i;
       until i > n−1
       Broadcasting_algorithm(Data, p, l+1, 0, 0, 0)
       /* p(i), k+1 ≤ i ≤ n−1, broadcast M in A^(l+1)_(n,k)(Elect_i(p)) */
    end;
/* end Phase 2 */
/* begin Phase 3: p(i), l+2 ≤ i ≤ k−1, broadcasts M in A^(i−1)_(n,k)(First_(i−1)(p)) */
    if ID = 1 then begin
       l := I; m := 0;
       repeat /* Phase 1 of the source node */
          m := m+1; i := k+1−2^(m−1);
          if i ≥ l+2 then p sends M(Data, p_out, l, i, m, 1) to pg_in
       until i < l+2
       m := 0;
       repeat /* the source node broadcasts M to p(i), k+1 ≤ i ≤ n−1 */
          m := m+1; i := k+1+(2^(m−1) −1)(k−l);
          if i ≤ n−1 then p send M(Data, p_out, l, i, m, 2) to pg_(l+1)i
       until i > n−1
    end
/* end Phase 3 */
end;
```

Figure 2 illustrates the broadcasting procedure for $A_{7,4}$ to be decomposed into its 7 subgraphs $A^1_{7,4}$, 6 subgraphs $A^2_{7,4}$, 5 subgraphs $A^3_{7,4}$ and 4 subgraphs $A^4_{7,4}$ based on the source node and the intermediate nodes. Let the source node 1234 have the given $p_{out} = 567$ and be denoted 1234(567). The source node 1234(567) broadcasts the message to 3 intermediate nodes $p(i)$, $2 \leq i \leq 4$, in $A^1_{7,4}(1)$ and the other 6 subgraphs $A^1_{7,4}(Elect_i(p))$, $2 \leq i \leq 7$, after completing the broadcasting procedure of **Phase 1** and **2** from 1234(567) to the other 6 subgraphs $A^1_{7,4}$. In $A^1_{7,4}(1)$, there are 3 intermediate node $p(2) = 1437(562)$, $p(3) = 1274(563)$ and $p(4) = 1237(564)$ that received the message. These 3 nodes are the source nodes of $A^1_{7,4}(1)$, $A^2_{7,4}(12)$ and $A^3_{7,4}(123)$ respectively . Since $p(2) \in A^1_{7,4}(14)$ and $p \in A^1_{7,4}(12)$, the source node $p(2) = 1437(562)$ in $A^1_{7,4}(1)$ only needs to broadcast the message to $A^2_{7,4}(13)$, $A^2_{7,4}(15)$, $A^2_{7,4}(16)$ and $A^2_{7,4}(17)$ in executing the broadcasting procedure of **Phase 3**. In the same way as $p(2) = 1437(562)$ does on $A^1_{7,4}(1)$, the source node $p(3) = 1274(563)$ in $A^2_{7,4}(12)$ only

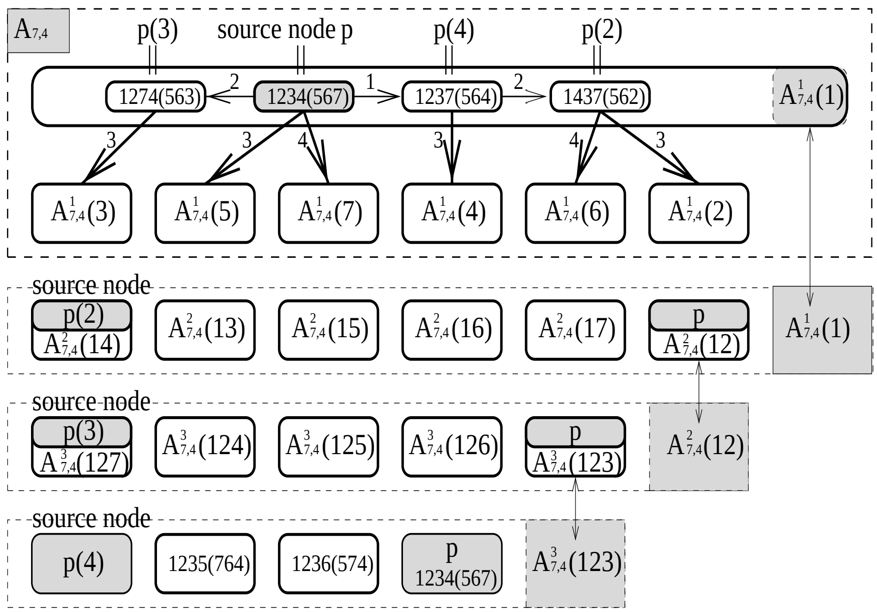

Figure 2: Partitioning $A_{7,4}$ into its subgraphs based on the properties of nodes

needs to broadcast the message to $A^3_{7,4}(124)$, $A^3_{7,4}(125)$ and $A^3_{7,4}(126)$, and the source node $p(4) = 1237(564)$ on $A^3_{7,4}(123)$ only needs to broadcast the message to $A^4_{7,4}(1235)$ and $A^4_{7,4}(1236)$. Since any one of $A^2_{7,4}$ can be considered as one $A_{5,2}$, we will take $A_{5,2}$ for example to illustrate *Broadcasting_algorithm* in figure 3.

Figure 3 illustrates the broadcasting tree from the source node 12 on $A_{5,2}$ by applying our algorithm. Let 12(345) be the source node. Each of the nodes that receive the message decides its p_{out} based on the received message. The number-labeled edges denote the steps in broadcasting. The node 15(342) is a intermediate node in **Phase 1**. It only sends to 13(542) and 14(352), and does not send to 12(345) on $A^1_{5,2}(1)$. This is because 12(345) holds the message and $A^2_{5,2}(12) = A^2_{5,2}(Elect_5(15(342)))$. This figure shows that our algorithm completes the broadcasting in 6 steps without redundant messages on $A_{5,2}$.

Now, we prove that the algorithm *Broadcasting_algorithm* is optimal in the time and message complexity.

Lemma 4 *After applying* **Phase 1** *and* **2**, *Broadcasting_algorithm can broadcast a message from the source node* p *in* $A^l_{n,k}(First_l(p))$ *to* $k-l-1$ *intermediate nodes in* $A^{l+1}_{n,k}(First_{l+1}(p))$ *and to only one node in each of* $n-l-1$ *node-disjoint*

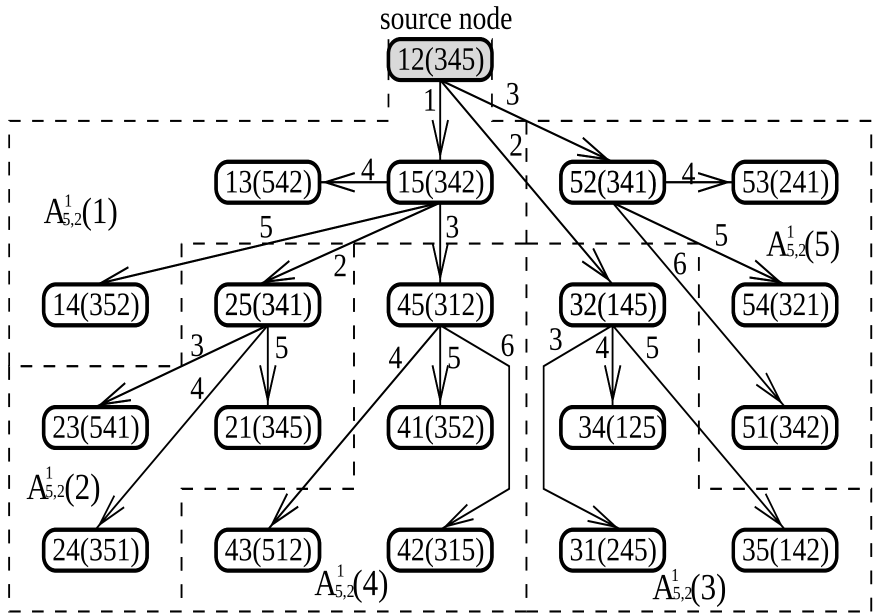

Figure 3: The broadcasting from the source node 12(345) in $A_{5,2}$

subgraphs $A^{l+1}_{n,k}(Elect_i(p))$*,* $l+2 \le i \le n$*, in* $O(\lg(n-l))$ *steps.*

Proof: In **Phase 1**, there are $k-l-1$ nodes intermediate $p(i)$, $l+2 \le i \le k$, that received the message. These nodes belong to $A^{l+1}_{n,k}(First_{l+1}(p))$. Since $k-l-1$ nodes $p(i)$, $l+2 \le i \le k$, send respectively the message to $k-l-1$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le k$, only once in **Phase 1**, and there is only one node $p(j)$ which received the message in **Phase 2** in each of $n-k$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_j(p))$, $k+1 \le j \le n$, the source node p can broadcast a message to only one node in each of $n-l-1$ node-disjoint subgraphs $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \le i \le n$.

Let L_1 and L_2 denote the number of the steps required for the nodes in **Phase 1** and **2** to broadcast the message to each of the other $n-l-1$ subgraphs $A^{l+1}_{n,k}$ only once. Let L denote the number of the steps required for p to complete the broadcasting procedure of **Phase 1** and **2**. As shown in **Phase 1**, since $l+2 \le k+1-2^{m-1}$ and $l+2 \le I-2^{m-1}$ with $max(I)=k$, $L_1 = max\{1+\lceil\lg(k-l-1)\rceil, 2+\lceil\lg(k-l-2)\rceil\} \le 2+\lceil\lg(k-l)\rceil$. Similarly, $L_2 \le 2+\lceil\lg\frac{n-l-2}{k-l}\rceil$ as shown in **Phase 2**. Therefore,

$$L = L_1 + L_2 = O(1) + O(\lg(k-l) + \lg\frac{n-l-2}{k-l}) = O(\lg(n-l)) \ .$$

□

Lemma 5 *In applying* **Phase 3***, each of* $k-l-1$ *intermediate nodes* $p(i)$*,* $l+2 \leq i \leq k$*, is the source node in each of* $k-l-1$ *different level subgraphs* $A^{i-1}_{n,k}(First_{i-1}(p))$*.*

Proof: Based on Lemma 2 and 3, it is trivial to prove Lemma 5. □

Theorem 2 *Broadcasting_algorithm on* $A^l_{n,k}$ *can broadcast a message from the source node to all the other* $\frac{(n-l)!}{(n-k)!}-1$ *nodes in at most* $O((k-l)\lg(n-l))$ *steps without redundant messages.*

Proof: We use induction to prove that *Broadcasting_algorithm* has no redundant messages on $A^l_{n,k}$. As a basis, it is easy to prove that this algorithm has no redundant messages for $l=k$ and $l=k-1$ on $A^l_{n,k}$ since $A^k_{n,k}$ is only a node and $A^{k-1}_{n,k}$ is a complete graph.

Assume that *Broadcasting_algorithm* has no redundant messages on $A^{l+1}_{n,k}$ for $1 \leq l+1 \leq k-2$.

Let p be the source node of broadcasting in $A^l_{n,k}(First_l(p))$. $A^l_{n,k}(First_l(p))$ can be decomposed into the $(l+1)$th level node-disjoint subgraphs as follows.

$$V^l_{n,k}(First_l(p)) = V^{l+1}_{n,k}(First_{l+1}(p)) \cup \{\bigcup_{l+2\leq i\leq n} V^{l+1}_{n,k}(Elect_i(p))\}$$

From Lemma 4, each of $A^{l+1}_{n,k}(Elect_i(p))$, $l+2 \leq i \leq n$, has a source node after applying **Phase 1** and **2**. Hence, *Broadcasting_algorithm* is no redundant messages on these subgraphs by the inductive hypothesis. From Lemma 5, the intermediate nodes $p(l+2)$ is the source node in the subgraph $A^{l+1}_{n,k}(First_{l+1}(p))$. Based on Lemma 2, $A^{l+1}_{n,k}(First_{l+1}(p))$ can be decomposed into the lower level node-disjoint subgraphs as follows.

$$V^{l+1}_{n,k}(First_{l+1}(p)) = \{\bigcup_{l+2\leq i\leq n-1} V^{l+2}_{n,k}(Elect_i(p(l+2)))\} \cup V^{l+2}_{n,k}(First_{l+2}(p)) \ .$$

Since the intermediate nodes $p(l+2)$ in **Phase 3** execute the broadcasting procedure of **Phase 1** and **2** except for not sending the message to $A^{l+2}_{n,k}(Elect_n(p(l+2))) = A^{l+2}_{n,k}(First_{l+2}(p))$, each of $A^{l+2}_{n,k}(Elect_i(p(l+2)))$, $l+3 \leq i \leq n-1$, has a source node after applying **Phase 3**. Since $p(l+2)$ is in $A^{l+2}_{n,k}(Elect_{l+2}(p(l+2)))$ and $p(i)$ for $l+3 \leq i \leq k$ is in $A^{l+2}_{n,k}(Elect_n(p(l+2))) = A^{l+2}_{n,k}(First_{l+2}(p))$, *Broadcasting_algorithm* has no redundant messages on $A^{l+1}_{n,k}(First_{l+1}(p))$ by inductive hypothesis. It is shown that our algorithm can broadcast the message from the source node to each of the other nodes exactly once on $A^l_{n,k}$.

Now, we prove that *Broadcasting_algorithm* takes $O((k-l)\lg(n-l))$ steps to complete broadcasting on $A^l_{n,k}$. Based on Lemma 4, our algorithm can reduce the broadcasting problem size on $A^l_{n,k}$ to the broadcasting problem size on $A^{l+1}_{n,k}$ in $O(\lg(n-l))$ steps for $0 \leq l \leq k-1$ recursively. Since, in the broadcasting procedure of our algorithm, $A^l_{n,k}$ is decomposed into its lower level subgraphs until $A^k_{n,k}$, the message is broadcast to all the other nodes on $A^l_{n,k}$ through at most $k-l$ source nodes in different lower level subgraphs. Let *Length* denote the number of the steps required for broadcasting a message on $A^l_{n,k}$.

$$Length \leq (k-l)O(\lg(n-l)) = O((k-l)\lg(n-l))$$

□

Theorem 3 *Broadcasting_algorithm can broadcast a message from the source node to all the other* $\frac{n!}{(n-k)!}-1$ *nodes in at most* $O(k\lg n)$ *steps without redundant messages on* $A_{n,k}$.

Proof: Based on Theorem 2, it is trivial to prove Theorem 3. □

As shown in Theorem 1, since an optimal broadcasting algorithm requires at most $\Omega(k\lg n)$ steps to complete broadcasting on $A_{n,k}$, *Broadcasting_algorithm* is optimal in the time complexity. Since this algorithm broadcasts a message only once to each of the other nodes on $A_{n,k}$, it is also optimal in the message complexity.

4 Conclusion

In this paper, we have presented a one-to-all distributed broadcasting algorithm for the fault-free arrangement graph interconnection network. This algorithm is based on the hierarchical property of the arrangement graph. It can broadcast a message to all the other nodes on the (n,k)-arrangement graph in at most $O(k\lg n)$ steps. It can also guarantee that each of the other nodes on the arrangement graph receives the message exactly once. It is shown that our *Broadcasting_algorithm* can perform the broadcasting on the arrangement graph with optimal time and message complexity.

Acknowledgments

The authors would like to thank the reviewers for their constructive and helpful comments. They have made many helpful revisions and suggestions that have significantly improved the presentation.

References

[1] S. B. Akers, D. Harel, and B. Krishnamuthy. The star graph: An attractive alternative to the n-cube. In *Proc. Int. Conf. Parallel Processing*, pages 393–400, 1987.

[2] S. B. Akers and B. Krishnamuthy. A Group-Theoretic Model for Symmetric Interconnection Networks. *IEEE Trans. Comput.*, 38(4):555–566, 1989.

[3] S. G. Akl, B. Qiu, and I. Stojmenovic. Fundamental algorithm for the star and pancake interconnection networks with applications to computational geometry. *Networks*, 23(4):215–225, 1993.

[4] L. Q. Bai, P. M. Yamakawa, H. Ebara, and H. Nakano. A Broadcasting Algorithm on the Arrangement Graph. *Computing and Combinatorics, Lecture Notes in Computer Science 959*, Springer, pages 462–471, 1995.

[5] K. Day and A. Tripathi. Arrangement graphs: A class of generalized star graphs. *Information Processing Letters*, 42:235–241, 1992.

[6] P. Fraigniaud. Asymptotically Optimal Broadcasting and Gossiping in Faulty Hypercube Multicomputers. *IEEE Trans. Comput.*, 41(11):1410–1419, 1992.

[7] E. Horowitz and A. Zorat. The binary tree as interconnection network: Application to multiprocessing systems and VLSI. *IEEE Trans. Comput.*. 30(4):245–253, 1981.

[8] S. L. Johnsson and C. T. Ho. Optimal Broadcasting and Personalized Communication in Hypercubes. *IEEE Trans. Comput.*, 38(9):1249–1268, 1989.

[9] V. E. Mendia and D. Sarkar. Optimal Broadcasting on the Star Graph *IEEE Trans. Parallel Distrib. Syst.*, 3(4):389–396, 1992.

[10] J. Misic and Z. Jovanovic. Communication Aspects of the Star Graph Interconnection Network. *IEEE Trans. Parallel Distrib. Syst.*, 7(5):678–687, 1994.

[11] J. P. Sheu, W. H. Liaw, and T. S. Chen. A broadcasting algorithm in star graph interconnection network. *Information Processing Letters*, 48:237–241, 1993.

[12] J. P. Sheu, C. T. Wu, and T. S. Chen. An Optimal Broadcasting Algorithm without Message Redundancy in Star Graphs. *IEEE Trans. Parallel Distrib Systs*, 6(6):653–658, 1995.

[13] J. Y. Tien, C. T. Ho, and W. P. Yang. Broadcasting on Incomplete Hypercubes. *IEEE Trans. Comput.*, 42(11):1393–1398, 1993.