アルゴリズム 41−12
（1994. 9. 21）

Optimal Broadcasting Algorithm on Arrangement Graph

Leqiang Bai † Peter Yamakawa †
Hiroyuki Ebara ‡ Hideo Nakano †

†Faculty of Engineering, Osaka University
‡Faculty of Engineering, Kansai University

A new interconnection network topology—the arrangement graph, as a generalization of the star graph topology, possesses excellent topology like the star graph and presents more flexibility than the star graph in terms of choosing the major design parameters: degree, diameter, and the number of nodes. In this paper, we propose an optimal distributed algorithm for one-to-all broadcasting on the *arrangement graph* in fault free mode. The algorithm, which exploits the rich inherent structure of the graph to constitute the structure of broadcasting binary tree and works recursively, broadcasts a message to all the $\frac{n!}{(n-k)!}$ processors in $O(klgn)$ steps.

arrangement graph についての最適な放送アルゴリズム

白 楽強 † Peter 山川 †
榎原 博之 ‡ 中野 秀男 †

† 大阪大学工学部通信工学科，吹田市
‡ 関西大学工学部情報処理教室，吹田市

arrangement graph は star graph を一般化したグラフであり、star graph 同様の良い性質を持っている。さらに、次数、直径、ノード数の点でより柔軟性のある設計ができる。この論文は arrangement graph について 最適な one-to-all 放送アルゴリズムを提案する。このアルゴリズムは グラフの階層的な構造を活用して再帰的に実行し、最適なステップ数 $O(klgn)$ ですべての $\frac{n!}{(n-k)!}$ プロセッサにメッセージ を放送することができる。

1 Introduction

Broadcasting is one of the communication problems for a distributed memory multicomputer. 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 need to be sent is called the destination of broadcasting. There are a large class of problems that cannot be solved in a timely fashion using today's sequential computers. But, many of these problems can be broken into smaller size problem that can be perform in the sequential computer in parallel. By broadcasting, each of these tasks can be assigned to a single processor and performed by this processor.

A widely studied interconnection network topology is the star graph[3]. It has been proposed as an attractive alternative to the hypercube with many superior characteristics. Particularly, for general purpose interconnection topology, where degree, distance and diameter are of primary concern, the star graph is clearly superior to the hypercube. A major practical difficulty with the star graph is related to its number of nodes: $n!$ for an n-star graph. The set of value of $n!$ spread widely over the set of integers. The arrangement graph is a new interconnection network topology. This topology brings a solution to the problem of growth of the number $n!$ of nodes in the $n-$star with respect to its dimension n. It also preserves all the nice qualities of the star graph topology such as: hierarchical structure, vertex and edge symmetry, simple shortest path routing and many fault tolerance properities.

The broadcasting problems on the hypercube and the stargraph have been investigated in recent years. In [4], Johnsson and Ho presented three new communication graphs for hypercube and defined scheduling disciplines. In [1], Mendia and Sarker proposed an optimal algorithm for one-to-all broadcasting in the star graph . It exploits the rich structure of the star graph and works by recursively partitioning the original the star graph into smaller star graphs.

In this paper, we consider the one-to-all broadcasting problem on the arrangement graph based on Mendia's work and propose an optimal broadcasting algorithm in the message passing model. By applying the ideas of binary tree into the rich topology of the arrangement graph, this algorithm performs the optimization on the time complexity. The remainder of this paper is organized as following. Interconnection network, arrangement graph and its properties are introduced in Section II. Our algorithm in $O(klgn)$ steps is proposed in Section III. We conclude the results in Section V.

2 Arrangement Graph

2.1 Definition

Let n and k, for $1 \leq k \leq n$, be two integers, and let us denote $\langle n \rangle = \{1, 2, \ldots, n\}$ and $\langle k \rangle = \{1, 2, \ldots, k\}$. Let $P_{n,k}$ be the set of permutations of the n elements of $\langle n \rangle$ taken k at a time. The k elements of an arrangement 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:*

$$V = \{p_1 p_2 \ldots p_k \mid p_i \ in \ \langle n \rangle \ and \ p_i \neq p_j \ for \ i \neq j\} = P_{n,k}, \ and$$

$$E = \{(p, q) \mid p \ and \ q \ in \ V \ and \ for \ some \ i \ in \ \langle k \rangle, \ p_i \neq q_i \ and \ p_j = q_j \ for \ j \neq i\}. \quad \Box$$

Fig.1 shows the arrangement graphs for the case $A_{4,2}$.

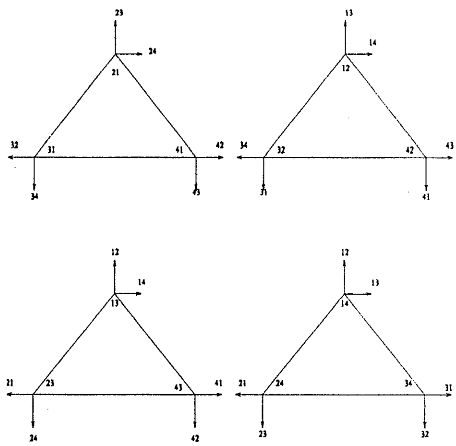

Figure 1: (4,2)-arrangement graph

2.2 Basic Properties

The (n,k)-arrangement graph is regular of degree $k(n-k)$, number of nodes $\frac{n!}{(n-k)!}$, and diameter $\lfloor \frac{3}{2} k \rfloor$. When designing an interconnection network based on the arrangement graph topology, we can, by tuning the two parameters n and k, make a more suitable choice for the number of nodes and for the degree/diameter tradeoff. The arrangement graph is a superset of many interconnection network topologies. For example, the arrangement graph $A_{n,n-1}$ is isomorphic to the n-star graph, and the arrangement graph $A_{n,1}$ is isomorphic to the complete graph with n nodes. The arrangement graph $A_{n,k}$ can be considered as a level-n hierarchical graph. Notice that there are $\frac{(n-1)!}{(n-k)!}$ nodes in $A_{n,k}$ which have element i in position j for any fixed i and j $(1 \leq i \leq n,\ 1 \leq j \leq k)$. These nodes form an $A_{n-1,k-1}$ subgraph of $A_{n,k}$ consisting of all nodes with element i in position j. For a fixed position j, the subgraphs $1_j,\ 2_j, \ldots, n_j$ have disjoint sets of nodes, therefore they form a partitioning of the set of nodes of $A_{n,k}$. This partitioning into n copies of $A_{n-1,k-1}$ can be done in k different ways and can be carried out recursively.

The arrangement graph still has many other good characteristics. For a more thorough coverage of the arrangement graph, the reader is referred to [2].

3 One-to-All Broadcasting

In the section II, we have introduced a new topology (arrangement graph) that presents more flexibility than the star graph in adjusting the major design parameters : numbers of nodes , degree and diameter. It is well-known that the broadcasting is the most important communication primitives in multiprocessors with no common memory. If the arrangement graph is considered the attractive candidate as a general architecture, it must have not only the significant properties of architecture and but also the efficient broadcasting algorithm.

3.1 Assumption and Theorem on Broadcasting

Based on the qualities of computer and network, we know that some node for some network can communicate with any nodes connected to it and some node can communicate with at most one other node connected to it at the given time. According to the relationship among nodes in broadcasting, there are one-to-all broadcasting and all-to-all broadcasting[4]. It means that the broadcasting has the different definition in the different condition. Here we consider one-to-all broadcasting. In this broadcasting model, we make the following assumption:

A node consists of processor with full duplex communication links to each of the other nodes connected to it. Any nodes have the enough buffer to preserve the received message. At the given time, a node can communicate with at most one other node connected or adjacent to it. There is no faults in the network.

According to this assumption, we can give a theorem about the lower bound of the time complexity for any one-to-all broadcasting algorithm.

Theorem 1 *:If an interconnection-network consists of N nodes or processors, then any one-to-all broadcasting algorithm on the network must take at least $\Omega(lgN)$ steps.*

Proof: As shown in the references [1] □

From above, it is known that a broadcasting algorithm is optimal on the time complexity if and only if it meets the condition given in Theorem 1. Our purpose is to look for such an optimal algorithm based on the properties of the arrangement graph.

3.2 Broadcasting in $O(kn)$

Let $p^0 = p_1 p_2 ... p_k$ be the node of the arrangement graph $A_{n,k}$,we define the set $INT(p^0) = \{p_1, p_2, ..., p_k\}$ of the k elements of $\langle n \rangle$ used in the node p^0, and the set $EXT(p^0) = \langle n \rangle - INT(p^0)$. The elements of $INT(p^0)$ is referred to as the internal elements of p^0 and the elements of $EXT(p^0)$ is referred to as the external elements of p^0.

From the properties of the arrangement graph, we know that the arrangement graph has hierarchical structure. Utilizing this structure, we will examine how to generate a broadcasting algorithm.

Notice that the graph $A_{n,k}$ contains n disjoint subgraphs $A_{n-1,k-1}$ with the elements $p_i \in \{p_i \mid p_i \in \langle n \rangle, 1 \leq i \leq n\}$ in the position $j \in \{j \mid 1 \leq j \leq k\}$. Let i_j denote the subgraph of $A_{n,k}$ with the element i in the position j. For the fixed j, the subgraph $1_j, 2_j, ..., n_j$ are n disjoint sets of nodes, therefore we can divide all nodes of $A_{n,k}$ into n parts and each part forms a subgraph $A_{n-1,k-1}$ of $A_{n,k}$. Now we assume that the node p^0 broadcasts a message. In order to send the message to at least one node of n disjoint $A_{n-1,k-1}$ with the element i in the position j in $A_{n,k}$, we need to deal with the set of $A_{n-1,k-1}$ in the following two cases:

(1) The case-A : $i_j \in \{i_j \mid i \in EXT(p)\}$

(2) The case-B : $i_j \in \{i_j \mid i \in INT(p)\}$.

Let $EXT(i_j) = \{i_j \mid i \in EXT(p^0)\}$ and $INT(i_j) = \{i_j \mid i \in INT(p^0)\}$. Let $P^j = \{p^1, p^2, \ldots, p^k\} = \{p_1 p_2 ... i ... p_k \mid i \in EXT(p^0)\}$ be the set of the nodes that have the exactly one different element i in the position j with the node p^0. In case-A, the node p^0 is directly connected to i_j because the node p^0 has exactly one different element with the node p^j and $p^j \in EXT(i_j)$. It means that the node p^0 can send directly the message to $i_j \in EXT(i_j)$. Let $\xrightarrow{S}$ denotes the sending of the message from one node to another. We have the procedure of case-A: $p^0 \xrightarrow{S} EXT(i_j)$. For example, let p^0 be a node of the graph $A_{9,6}$ and $p^0 = 123456$. For $j = 6$, $EXT(i_6) = \{7_6, 8_6, 9_6\}$, the node p^0 is directly connected to each of the nodes in the set $\{12345\underline{7}, 12345\underline{8}, 12345\underline{9}\}$ because the node p^0 has only one different element with the nodes in the set $\{12345\underline{7}, 12345\underline{8}, 12345\underline{9}\} \in P^j$, and can send the message to the any nodes of the set P^j.

In case-B, the node p^0 is not directly connected to the any nodes belong to the set $i_j \in INT(p^0)$ except itself. So in order to send the message to the subgraph $i_j \in INT(i_j)$, we must send the message to some nodes that are connected not only to the node p^0 but also to the nodes belong to the subgraph $i_j \in INT(i_j)$. We will show that it is possible to find this kind of the nodes by exploiting the topology properties of the arrangement graph. From the definition, we know that the node p^0 is connected to the nodes belong to the set P^j and some of these nodes are connected to the some nodes belong to the set $i_j \in INT(i_j)$. This procedure can be presented as $p^0 \xrightarrow{S} P^j \xrightarrow{S} INT(i_j)$. For example, let $p^j = p_1p_2...p_n...p_k$ be a node that belongs to the subset P_n^j of P^j, which has the element $p_n \in EXT(p^0)$ in the position j. We can get k such nodes that are connected to each $i_j \in INT(i_j)$ respectively. We have

$$p^1 = \underline{p_n}p_2p_3...p_j...p_k$$
$$..................$$
$$p^j = p_1p_2p_3...\underline{p_n}...p_k$$
$$..................$$
$$p^k = p_1p_2p_3...p_j...\underline{p_n}$$

and the node p^0 is connected to any node $p^j \in \{p^j \mid 1 \le j \le k\}$. So the node p^0 can send the message to any node $p^j \in \{p^j \mid 1 \le j \le k\}$. From the relationship between the node p^j and the node of $i_j \in INT(i_j)$, we know that the node p^j can directly send the message to the node p_{ij} with the element $p_i \in INT(p^0)$ in the position j, which belongs to the subgraph $i_j \in INT(i_j)$. Consequently, we can use the following procedure to broadcast the message from the source p^0 to at least one node that belongs to the subgraph $i_j \in INT(i_j)$. We have

$$p^0 \xrightarrow{S} \begin{cases} p^1 \xrightarrow{S} p_{1j} \\ p^2 \xrightarrow{S} p_{2j} \\ p^3 \xrightarrow{S} p_{3j} \\ .... \\ p^i \xrightarrow{S} p_{ij} \\ .... \\ p^k \xrightarrow{S} p_{kj} \end{cases}$$

It means that this procedure can distribute the message to at least one node in each of $i_j \in INT(i_j)$ through some mediate node between the source node and destination node. Consequently combining the procedures of the case-A and the case-B, we can correctly distribute the message from the source node to at least one node in each of n distinct $A_{n-1,k-1}$ comprising the original $A_{n,k}$.

Lemma 1 :*Given an arrangement graph $A_{n,k}$ in which there is one node containing a message to be broadcasted, the procedures in the case-A and the case-B along with a sequential algorithm to be performed by every node as described above will correctly distributed the message to at least one node in each of n distinct $A_{n-1,k-1}$ comprising the original $A_{n,k}$.*

Proof :

$$\forall \ \textit{source node} \ p^0$$
$$\exists \ EXT(i_j) \ \textit{and} \ INT(i_j)$$
$$\textit{Since} \ p^0 \xrightarrow{S} \{EXT(i_j) \cup P^j\},$$
$$\textit{and} \ P^j \xrightarrow{S} INT(i_j)$$
$$\textit{then} \ p^0 \xrightarrow{S} \{EXT(i_j) \cup INT(i_j)\}$$
$$\textit{and} \ \{EXT(i_j) \cup INT(i_j)\} \overset{S}{\Leftrightarrow} \{i_j \mid i \in \langle n \rangle\}$$
$$\textit{Hence} \ p^0 \xrightarrow{S} \{i_j \mid i \in \langle n \rangle\}$$

So far, we have shown that the message can correctly distributed to each of n distinct $A_{n-1,k-1}$ comprising the original $A_{n,k}$. □

We know that the arrangement $A_{n,k}$ is a level-k hierarchical graph. From the Lemma 1, a message from the source node can be correctly distributed to each of n distinct A_{n-1k-1}, In the same way, we can distribute a message from level-j graph to level-$(j-1)$ for $1 \le j \le k$. If we apply the technique recursively into each subgraph until the problem is reduced to broadcasting on $A_{n-(k-1),1}$, we would distribute the message to every node of $A_{n,k}$. At this point, we will guarantee that the broadcasting is finished and every node has received the message.

For convenience, we define the some notations. In $A_{n,k}$, let the set of $p^0 \xrightarrow{S} \{p_1p_2...p_j...p_k \mid p_j \in EXT(i_j)\}$ be denoted by DPC_I_k , the set of $p^0 \xrightarrow{S} \{p^j \mid 1 \le j \le k\}$ be denoted by PPC_II_k and the set of $\{p^j \mid 1 \le j \le k\} \xrightarrow{S} \{p_1p_2...p_j...p_k \mid p_j \in INT(i_j)\}$ be denoted by DPC_II_k. In the same way, we can define the sets $PPC\_I_{k-j}, PPC\_II_{k-j}, DPC\_II_{k-j}$ in the subgraph $A_{n-j,k-j}$ for $1 \le j \le k-1$. For the subgraph $A_{n-(k-1),1}$, we have only DPC_I_1 because of no $INT(i_j)$ except the source node itself.

Theorem 2 : *For an arrangement $A_{n,k}$, the procedure sequence $BS\_A_{n,k}$ of the form:*

$$BS\_A_{n,k} = \{DPC\_I_k, \ PPC\_II_k, \ DPC\_II_k,$$
$$DPC\_I_{k-1}, PPC\_II_{k-1}, DPC\_II_{k-1},$$
$$DPC\_I_{k-2}, PPC\_II_{k-2}, DPC\_II_{k-2},$$
$$..$$
$$DPC\_I_{k-j}, PPC\_II_{k-j}, DPC\_II_{k-j},$$
$$..$$

DPC_I_3, PPC_II_3, DPC_II_3, DPC_I_2, PPC_II_2, DPC_II_2, DPC_I_1.}

along with a sequential algorithm performed by every node described above constitutes a broadcasting algorithm.

Proof : We prove by recurrence that the algorithm constituted by this sequence will correctly broadcast a message to all of the nodes $P(n,k)$ in $A_{n,k}$. Let a node p^0 in $A_{n,k}$ has a message to be broadcasted. By the distributing sequence $\{DPC\_I_k, PPC\_II_k, DPC\_II_k\}$, the node p^0 can distribute the message to at least node of each of n distinct $A_{n-1,k-1}$ and some mediate nodes p^j. Let P_{k-j} denote the set of $n-j$ nodes consisting of exactly one node of each of $n-j$ distinct $A_{n-(j-1),k-(j-1)}$, in which the nodes have received the message by the procedure sequence $\{DPC\_I_{k-j}, PPC\_II_{k-j}, DPC\_II_{k-j}\}$. We have $|P_{k-j}| = n-j$. It means that each of the nodes in $n-j$ distinct $A_{n-(j-1),k-(j-1)}$, which comprise $A_{n-j,k-j}$, has received the message. From the structure of the procedure $BS\_A_{n,k}$, we know that this sequence can be broken into k stages. Utilizing the rule of product, the procedure sequence $BS\_A_{n,k}$ will guarantee that at least

$$\prod_{j=0}^{k-1} |P_{k-j}| = \prod_{j=0}^{k-1}(n-j) = P(n,k)$$

distinct nodes would receive the message. Since $A_{n,k}$ have exactly $P(n,k)$ nodes, it will be sure that every node would receive the message after performing the sequence $BS\_A_{n,k}$. □

3.3 Analysis of broadcasting

According to the features of the parallel communication and the definition of the procedures DPC_I_{k-j}, PPC_II_{k-j} and DPC_II_{k-j}, we can calculate the number of steps of broadcasting a message from a node to all other nodes in an arrangement $A_{n,k}$. Let the total length of the sequence be L

$$\begin{aligned} L &= \sum_{j=0}^{k-1}(DPC\_I_{k-j} + PPC\_II_{k-j} + DPC\_II_{k-j}) \\ &= \sum_{j=01}^{k-1}(n-k) + \sum_{j=0}^{k-2}(k-j) + \sum_{j=0}^{k-2} 1 \\ &= O(nk) \end{aligned}$$

As indicated above, we have shown that the sequence $BS\_A_{n,k}$ can constitute a broadcasting algorithm and its time complexity is $O(nk)$.
In an arrangement $A_{n,k}$, there are $P(n,k)$ nodes. As described in Theorem 1, an optimal algorithm requires at least $O(lg(P(n,k)) = O(lg\frac{n!}{(n-k)!}) = O(klg(n))$ steps. Therefore the algorithm performed by the sequence $BS\_A_{n,k}$ is not optimal. For solving this problem, we use of the concept of the binary tree. If we can embed the procedure $BS\_A_{n,k}$ into a binary tree, we could produce an broadcasting algorithm in $O(klgn)$.
It is known that an arrangement $A_{n,k}$ has k levels hierarchical structure. The sequence $BS\_A_{n,k}$ can be also divided into k levels and the subsequence of each level can be used to distribute the message from higher level graph to next lower level graph. From the structure of the subsequence, each subsequence except DPC_I_1 has three parts, i.e., DPC_I, PPC_II and DPC_II. For some $A_{n-j,k-j}$, the length of DPC_I_{k-j} and PPC_II_{k-j} is $n-k$ and k respectively. However the length of DPC_II_{k-j} is always one . If we are to reduce the time complexity of the algorithm in $O(kn)$, it must be by reducing the time complexity of the sequence DPC_I_{k-j} and PPC_II_{k-j}. In fact, if the nodes receiving the message by the sequence DPC_I_{k-j} and the sequence PPC_II_{k-j} were embedded on a binary tree, then we could easily develop an algorithm to perform the sequence DPC_I_{k-j} , PPC_II_{k-j} and DPC_II_{k-j} in $O(klgn)$.

3.4 Broadcasting in $O(klgn)$

We assume the node $p^0 = p_1p_2...p_k$ belongs to the graph $A_{n,k}$. The node p^0 corresponds to an arrangement of k elements chosen out of the n elements. Let the set of the positions of the elements of the arrangement be the $J_{in} = \{1,2,...,k\}$ and the set of the virtual positions of the external elements of the arrangement is $J_{out} = \{k+1, k+2, ..., n\}$, then we can define a bijective function $f: INT(p^0) \cup EXT(p^0) \rightarrow J_{in} \cup J_{out}$ with $f = \{(p_1,1),...,(p_k,k),(p_{k+1},k+1),...,(p_n,n)\}$, i.e., $f: INT(p^0) \rightarrow J_{in}$ with $f = \{(p_1,1),...,(p_k,k)\}$ and $f: EXT(p^0) \rightarrow J_{out}$ with $f = \{(p_{k+1},k+1),...,(p_n,n)\}$.
Let a node p^0 have a message to be broadcasted. Every node can decide whether or not to send the message to some other nodes. If we can design a kind of numbered method based on the message so that every node receiving the message can be embedded into a binary tree, an optimal broadcasting could be performed. As described in the algorithm in $O(kn)$, the node p^0 can be connected to any node in the set P^j. If we properly adjust the elements of the nodes in the set P^j so that a kind of the relation of binary tree exists among the node p^0 and the nodes in the set P^j, the sequence PPC_II_k can be performed in the $\lceil lg(k+1)\rceil$ steps.
As shown in Table 1, let $\xrightarrow{m}$ denote the sending of the message at the m-th step. First let

the node p^0 send the message to the node $p^1 = \underline{p_n}p_2p_3...p_k$. At first step, we have $f = \{\underline{(p_n, 1)},..., \underline{(p_1, n)}\}$. Note the element $p_n \in INT(p^0)$ and the element $p_1 \in EXT(p^0)$ for the node p^1. The other elements of p^1 are the same as these of p^0. At the second step, the node p^0 and p^1 could send the message to the adjacent to it respectively, i.e., $p^0 \xrightarrow{2} p^2$ and $p^1 \xrightarrow{2} p^3$. In the same way, an additional step will double the number of nodes that have received the message. At least $\lceil lg(k+1) \rceil$ steps are necessary for the node p^0 to distribute a message to some of the mediate nodes between the node p^0 and the nodes of $i_j \in INT(i_j)$. It means that the procedure of Table 1 provides a method to perform the sequence PPC_II_k in the $O(lgk)$ steps. From the definition of p^j, we know that p^j and p_{ij} have only one different element p_j in the position j. So after the end of the Table 1, the mediate nodes $p^j \in P_j$, which have received the message, can directly send the message to the subgraph $i_j \in INT(i_j)$ in the same step. Utilizing the Table 1 and the procedure DPC_II_k, we could distribute the message to at least one node in each of k distinct $i_j \in INT(i_j)$ at the step $\lceil lg(k+1) \rceil + 1$.

Notice that the element p_n is the external element for the node p^0. In the processes given by Table 1 and DPC_II_k, we use only the external element p_n of the node p^0 to produce a series of the mediate nodes and use these nodes to connect with the subgraph $i_j \in INT(i_j)$ for the node p^0. From this point, we find that the elements of the set $EXT(p^0)$ are still the external elements except the element p_n for all of the nodes p^j and the nodes of $INT(i_j)$,which have received the message. According to the definition of the arrangement graph, the nodes of P^j and $INT(i_j)$ that are produced by Table 1 and DPC_II_k are connected with some nodes of $EXT(i_j)$ for the node p^0. So we can utilize these nodes to distribute the message to some nodes that belong to the $EXT(i_j)$. Applying the concept of the binary tree into this process, we can establish Table 2 to perform this procedure of distributing the message.

In the Table 2, let the p^y denote some mediate node in order to send the message to the node of some subgraph $A_{n-1,k-1}$, and the p_{ij} denote the node of the subgraph $A_{n-1,k-1}$ that has the element p_i in the fixed position j. Let $\xrightarrow{m}$ denote the sending of the message at the m-th step. Initially there are k mediate nodes to send the message to the adjacent k nodes p_{ij} in the k subgraphs $A_{n-1,k-1}$. As shown in Table 2, an additional step doubles the number of nodes that have received the message. Until the m steps, the $k(2^m - 1)$ nodes will receive the message. We know that there are n distinct subgraph i_j in $A_{n,k}$ and at least one of them has one node having the message. For distributing the message to the remaining exactly $n-1$ subgraphs i_j, we set the constraint condition in Table 2 to control the distributing procedure. In this way, a message would be distributed to the $n-1$ distinct subgraphs in the optimal steps and the distributing procedure would end automatically according to the constraint condition.

Theorem 3 : *Let p^0 be a node of the arrangement graph $A_{n,k}$ with a message to be broadcasted. If the procedures of Table 1 and Table 2 are performed in order of the number of Table by every node as described above, the message can be correctly distributed to at least one node in each of n distinct subgraph $A_{n-1,k-1}$ comprising the original arrangement $A_{n,k}$ in $O(lgn)$.*

Because of the sake of the space, we omit the proof of Theorem 3. It would be proved by induction through analyzing the relation between the arrangement of node, which has received the message, and the arrangement of the source node. From the Theorem 3, it is known that the procedure of PPC_II_k, DPC_II_k and DPC_I_k can be performed in order in $O(lgn)$ steps. We will now present an algorithm to accomplish the procedure as described in Theorem 3.

```
procedure Broadcast
  (var M : message; DATA : {M, i, j, m});
  var i, j, m : integer;
  const L1 = ⌈lgk⌉; L2 = ⌈lg (n-1+k)/k⌉;
begin
  *Phase 1(For preparing)
  begin
    if node is source then
      begin
        for m := 1 to L1 do
          j := 2^(m-1);
        if j ≤ k then
          p^0 sends DATA to p^j
      end
    else
      if message received then
        x := j; m1 := m + 1;
        begin
          for m := m1 to L2 do
            j := 2^(m-1) + x;
          if j ≤ k then
            p^x sends DATA to p^j
  end{Phase 1 turn to Phase 2};
  *Phase 2(For broadcasting)
  begin
    if node has message then
      begin
        for m := 1 to L2 do
          i := k(2^(m-1) - 1) + j;
```

```
        if i ≤ n − 1 then
          p^j sends DATA to p_i1
      end
    else
      if message received then
        y := j; m_2 := m + 1;
        begin
          for m := m_2 then L_2 do
            i := k2^(m−1) + y;
          if i ≤ n − 1 then
            p_y1 sends DATA to p_i1
end{broadcast}
```

From this algorithm, we can observe that any nodes not participating in broadcasting procedure can work on their own tasks without knowledge of the broadcast. So, it can be considered both as synchronous algorithm and as asynchronous algorithm. By Theorem 3, the time complexity of this algorithm is therefore $O(lgn)$. Since this algorithm would be applied recursively k times as shown in Theorem 2, the time complexity of the entire broadcasting on $A_{n,k}$ would be $O(klgn)$.

4 Conclusion

The arrangement graph topology has many good topology properties and is an attractive architecture for interconnection networks. In the interconnection network, the problem how to broadcast a message from a node to all other nodes in the shortest steps has be paid a lot of attention, because broadcasting algorithm constitutes one of the most important communication primitives used in interconnection networks with distributed memory.

In this paper, we have presented an optimal distributed broadcasting algorithms for the arrangement graph in fault free mode. At first, by using the hierarchical property of the arrangement graph to partition a higher order $A_{n,k}$ into its component subgraph $A_{n-1,k-1}$, we have generated a broadcasting procedure in $O(kn)$ steps. We then utilized the inherent structure of the arrangement graph to embed in order the procedures of PPC_II_k, DPC_II_k and DPC_I_k into the binary tree so that a message would be distributed from one node in $A_{n,k}$ to at least one node in each of $A_{n-1,k-1}$ comprising the original $A_{n,k}$ in $O(lgn)$ steps. Finally by exploiting hierarchical structure of $A_{n,k}$, we have shown that it would be possible to perform an optimal one-to-all broadcasting on $A_{n,k}$ in $O(klgn)$ steps.

References

[1] Victor E. Mendia and Dilip Sarker, "Optimal Broadcasting on the Star Graph," *IEEE Trans. Parallel Distributed Syst.*, vol.3, no. 4, pp. 389-396, 1992.

[2] Khaled Day and Anand Tripathi, "Arrangement graphs : A class of generalized star graph," *Information Processing Letters* 42(1992)235-241, North-Holland, 3 July 1992.

[3] S. B. Akers, D. Harel, and B. Krishnamurthy, "The star graph: An attractive alternative to the n-star," in *Proc. Int. Conf. Parallel Processing,* 1987, pp. 393-400.

[4] S.Lennart Johnsson and Ching-Tien Ho, "Optimum Broadcasting and Personalized Communication in Hypercube," *IEEE Trans. Comput.*, vol. 38, no. 9, pp. 1249-1269, 1989.

[5] E. Horowitz and A. Zorat, "The binary tree as interconnection network: Applications to multiprocessing systems and VLSI," *IEEE Trans. Comput.*, vol. C-30, pp. 247-253. 1981.

[6] Pireer Fraigniaud, "Asymptotically Optimal Broadcasting and Gossiping in Faulty Hypercube Multicomputers," *IEEE Trans. Comput.*, vol. 41, no. 11, pp. 1410-1419, 1992.

[7] Jenn-Yang Tien, Ching-Tien Ho, and Wei-Pang Yang, "Broadcasting on Incomplete Hypercube," *IEEE Trans, Comput,*. vol. 42, no. 11, pp. 1393-1398, 1993

[8] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, *Introduction to Algorithms.* Cambridge, The MIT Press, 1990.

Table 1: Preparing Procedure

Sending	Node receiving message	f	Logical Relation
p^0	$p^0 = p_1 p_2 \ldots\ldots\ldots\ldots p_k$	(p_n, n)	$p_n \in EXT(p^0)$
$p^0 \xrightarrow{1} p^1$	$p^1 = \underline{p_n} \ldots\ldots\ldots\ldots p_k$	(p_1, n)	$p_1 \in EXT(p^0)$
$p^0 \xrightarrow{2} p^2$	$p^2 = p_1 \underline{p_n} \ldots\ldots\ldots\ldots p_k$	(p_2, n)	$p_2 \in EXT(p^0)$
$p^1 \xrightarrow{2} p^3$	$p^3 = p_n p_2 \underline{p_1} \ldots\ldots\ldots\ldots p_k$	(p_3, n)	$p_3 \in EXT(p^0)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$p^x \xrightarrow{m} p^j$	$p^j = \{\text{——}p^x\text{——}\}\underline{p_j} \cdots p_k$ $j = x + 2^{m-1}$ **Contraints Condition :** **END if $j > k$** $p_j = p_n$ if $x = 0$ $p_j = p_x$ if $x \neq 0$	(p_j, n)	$p_j \in EXT(p^0)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$p^x \xrightarrow{L_1} p^k$	$L_1 = \lceil lg(k+1) \rceil$ $x = k - 2^{L_1 - 1}$ $p_k = p_n$ if $x = 0$ $p_k = p_x$ if $x \neq 0$ $p^k = \{\text{———}p^x\text{———}\}\underline{p_k}$	(p_k, n)	$p_k \in EXT(p^0)$

Table 2: Broadcasting Procedure

Source Nodes	Send Message(M) to	Destination Nodes	Nodes Having M
$\{p^1, \ldots, p^k\}$	$\xrightarrow{1}$	$\{p_{1j}, \ldots, p_{kj}\}$	$p_{1j}, \ldots, p_{kj}$
$\{p^1, \ldots, p^k\}$	$\xrightarrow{2}$	$\{p_{(k+1)j}, \ldots, p_{(2k)j}\}$	
$\{p_{1j}, \ldots, p_{kj}\}$	$\xrightarrow{2}$	$\{p_{(2k+1)j}, \ldots, p_{(3k)j}\}$	$p_{1j}, \ldots, p_{(3k)j}$
	$\vdots$		$\vdots$
$p^y \xrightarrow{if} p_{ij} \Rightarrow$	$i = (2^{m-1} - 1)k + y \Rightarrow$	**Contriants Condition:**	
$p_{yj} \xrightarrow{if} p_{ij} \Rightarrow$	$i = k2^{m-1} + y \Rightarrow$	**END if $i \geq n$**	
p^1	$\xrightarrow{m}$	$p_{((2^{m-1}-1)k+1)j}$	
	$\vdots$		
p^y	$\xrightarrow{m}$	$p_{((2^{m-1}-1)k+y)j}$	
p_{yj}	$\xrightarrow{m}$	$p_{(k2^{m-1}+y)j}$	
	$\vdots$		
$p_{(k2^{m-1}-k)j}$	$\xrightarrow{m}$	$p_{(k2^m-k)j}$	$p_{1j}, \ldots, p_{(k2^m-k)j}$
	$\vdots$		$\vdots$
	$L_2 = \lceil lg \frac{n-1+k}{k} \rceil$		
$\{p^1, \ldots, p^y\}$	$p^y \xrightarrow{if} p_{(n-1)j}$ $y = n - 1 - k(2^{L_2-1} - 1)$ $\xrightarrow{L_2}$	$\{p_{(n-y)j}, \ldots, p_{(n-1)j}\}$	END $p_{1j}, \ldots, p_{(n-1)j}$
$\{p^1, \ldots, p^k\}$	$p_{yj} \xrightarrow{if} p_{(n-1)j}$ $y = n - 1 - k2^{L_2-1}$ $\xrightarrow{L_2}$	$\{p_{((2^{L_2-1}-1)k+1)j}$ $, \ldots ,$ $\{p_{((2^{L_2-1}-1)k+k)j}\}$	
	$\vdots$		END
$\{\ldots, p_{yj}\}$	$\beta := \lfloor \frac{n-1-k2^{L_2-1}}{k} \rfloor$ $\xrightarrow{L_2}$	$\{\ldots, p_{(n-1)j}\}$	$p_{1j}, \ldots, p_{(n-1)j}$