Permutation Routing and Gossiping on the Graph G(n,k) of the Johnson Scheme

Jaime Caro and Hans Riyono Ho
Department of Computer Science
University of the Philippines
Diliman, Quezon City, Philippines
jaime.caro@up.edu.ph

ABSTRACT

The graph $G(n,k)$ of the Johnson Scheme which is also known as the "Slice of the Cube" is the undirected graph where the vertices are all the k-subsets of the fixed n-set and such that two vertices A and B are adjacent if and only if $\mid A \cap B \mid = k-1$.

In this paper, we shall provide algorithms for one-to-one permutation routing, gossiping, and sorting on $G(n,k)$ that runs in $O(nk^2)$, $O(min(k, n-k))$ and $O(N)$, respectively, where N is the number of vertices.

1. INTRODUCTION

In multicomputer interconnection, static networks are used frequently because they have an important advantage: the degree of a node either remains fixed regardless of the size of the network or grows very slowly with network size. This allows very large networks to be constructed.

This gives us the motivation to study the graph $G(n,k)$ of the Johnson Scheme as a static interconnection network topology. By studying this graph, we hope to contribute some information regarding the efficiency and effectiveness of this graph as an interconnection network.

Some properties of the graph $G(n,k)$ of the Johnson Scheme have been studied such as the hamiltonicity, the diameter, connectivity and wide-diameter of the graph. There are some other properties related to the graph $G(n,k)$ of the Johnson Scheme that are of interest in interconnection network design theory. In particular, the algorithms for permutation routing, gossiping and sorting on the graph $G(n,k)$ of the Johnson Scheme are not yet studied.

Packet routing is the problem of sending a packet from a vertex to another vertex of a graph with the only restriction that at most one packet can traverse an edge at a time.

The basic permutation routing problem is defined on a graph as follows: for all i, vertex v_i wants to send a packet to vertex $d(v_i)$, where d is a permutation of the vertices of the graph.

One-to-one permutation routing of a graph is defined as follows: every vertex initially sends at most one packet and is the final destination of at most one packet.

In a graph, if a vertex has more than one packet which will go out using the same edge, then the packets can be stored at a vertex until they can use the edge.

In a graph, if a vertex v_i wants to send a packet p_i to a vertex $d(v_i)$, given that the vertex v_i's and vertex $d(v_i)$'s addresses are a set of numbers in ascending order, then scan the numbers of $d(v_i)$ from left to right, and compare them with the address of the current location of p_i. Send p_i out of the current vertex along the edge corresponding to the left-most numbers in which the current position and $d(v_i)$ differ.

One-to-all broadcasting is an operation where a vertex (source) must send a packet to all other vertices of the graph.

All-to-all broadcasting, also known as gossiping or total exchange, is a generalization of one-to-all broadcasting in which all vertices simultaneously initiate a broadcast. In other words, gossiping is the process whereby each vertex sends a packet to all the others.

Gossiping is done as follows: Let $G = (V, E)$ be a graph. With each vertex v, associate an initial packet. Each vertex can send its packet to a neighbor or neighbors and/or receive a packet from a neighbor or neighbors depending on the model of communication used. After receiving any packets, vertices take the union of all packets received at that step, thus forming new packets for the next step. At the other steps, when we take the union of all packets, we disregard multiplicities of packets.

Full-duplex communication model means that a vertex can simultaneously send and receive on all its edges at the same time. Half-duplex communication model means that a vertex can either send or receive on all its edges at a time.

2. THE GRAPH $G(N,K)$ OF THE JOHNSON SCHEME

The graph $G(n,k)$ of the Johnson Scheme is the undirected graph where the vertices are all the k-subsets of a fixed n-set. Two vertices A and B are adjacent if and only if $\mid A \cap B \mid = k-1$. We shall assume that if $\{a_1, a_2, a_3, \ldots, a_n\}$ denotes the elements of a vertex, then $a_1 < a_2 < a_3 < \cdots < a_n$. We know that the total number of vertices is $\binom{n}{k}$. Each vertex will have $nk - k^2$ edges incident to it, or in other words, the graph is $nk - k^2$ regular.

Consider the order of the elements in a vertex, we make a consensus that the order of the elements in a vertex is in non-decreasing order. By this consensus, we divide the graph $G(n, k)$ of the Johnson Scheme into subgraphs S_i such that each subgraph S_i has vertices with a common first element, a_1. In general, the graph $G(n, k)$ of the Johnson Scheme will have $n - k + 1$ such subgraphs, since the first element in the last subgraph is $(n - k) + 1$.

Some properties, such as the diameter, connectivity and wide-diameter, of graph $G(n, k)$ of the Johnson Scheme were proven by Muga II, Caro, Adorna, and Baes in [4]. The diameter of $G(n, k)$ was shown to be $min(k, n-k)$. It was also shown that the connectivity of graph $G(n, k)$ of the Johnson Scheme is $nk - k^2$. Finally, the wide-diameter, $d_\kappa(G)$, of $G(n, k)$ was shown to be $k + 1$.

In [2], it has been proven that the graph $G(n, k)$ is Hamiltonian. We shall give another proof of this. After that, we shall state algorithms for one-to-one permutation routing, gossiping and sorting on the graph $G(n, k)$ of the Johnson Scheme.

3. HAMILTONICITY OF THE GRAPH $G(N, K)$ OF THE JOHNSON SCHEME

We shall determine whether the graph $G(n, k)$ of the Johnson Scheme is hamiltonian.

LEMMA 1. *Each subgraph S_i of the graph $G(n, k)$ of the Johnson Scheme has a hamiltonian path.*

Proof.

Consider subraph S_i of $G(n, k)$. For any $A \in V$ where V is the vertex set of $G(n, k)$, assume that $A = \{a_1, a_2, a_3, \ldots, a_k\}$, where a_i is ith element in S_i, for $i = 1, 2, 3, \ldots, n$. We also assume without loss of generality that $a_1 < a_2 < a_3 < \cdots < a_n$.

Thus, each subgraph S_i has a general form as follows:

$$\begin{aligned}
S_i = \{ & \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-1)}, a_k\}, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-1)}, a_{(k+1)}\}, \\
& \ldots, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-1)}, a_n\}, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_k, a_n\}, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_k, a_{(n-1)}\}, \\
& \ldots, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_k, a_{(k+1)}\}, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_{(k+1)}, a_{(k+2)}\}, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_{(k+1)}, a_{(k+3)}\}, \\
& \ldots, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-2)}, a_{(k+1)}, a_n\}, \\
& \ldots, \\
& \{a_i, a_{(i+1)}, a_{(i+2)}, \ldots, a_{(k-3)}, a_{(k+1)}, a_{(k+2)}\}, \\
& \ldots, \\
& \{a_i, a_{(i+2)}, a_{(i+3)}, \ldots, a_k, a_{(k+1)}\}, \\
& \{a_i, a_{(i+2)}, a_{(i+3)}, \ldots, a_k, a_{(k+2)}\}, \\
& \ldots, \\
& \{a_i, a_{(i+2)}, a_{(i+3)}, \ldots, a_k, a_n\}, \\
& \ldots, \\
& \{a_i, a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}\}
\end{aligned}$$

for $i = 1, 2, \ldots, n - k + 1$.

We can examine the form and claim that two consecutive vertices are adjacent. Because the intersection of both is $k - 1$. Thus, there is hamiltonian path in every subgraph of the graph $G(n, k)$ of the Johnson Scheme.

LEMMA 2. *The first vertex of subgraph S_i is adjacent to the first vertex of subgraph $S_{(i+1)}$.*

Proof.

Let A_1 be the first vertex of subgraph S_1 and B_1 is the first vertex of subgraph S_2. It follows that

$A_1 = \{1, 2, 3, \ldots, k\}$,

$B_1 = \{2, 3, 4, \ldots, k + 1\}$,

So, the intersection between A_1 and B_1 is $\{2, 3, 4, \ldots, k\}$ which has cardinality $k - 1$. Thus, they differ in only one element. For A_1, it differs one element Thus, A_1 is adjacent to B_1. Let C_1 be the first vertex of subgraph S_3, that is $C_1 = \{3, 4, 5, \ldots, k + 2\}$. The intersection between B_1 and C_1 is $\{3, 4, 5, \ldots, k + 1\}$ which has cardinality $k - 1$. So, B_1 is adjacent to C_1. This adjacency also occurs between C_1 and D_1, D_1 and E_1, and so on, for D_1 is the first vertex of subgraph S_4, and E_1 is the first vertex of subgraph S_5. In fact, the first vertex of subgraph S_i is always adjacent to the first vertex of subgraph $S_{(i+1)}$ of the graph $G(n, k)$ of the Johnson Scheme.

LEMMA 3. *The last vertex of subgraph S_i is adjacent to the last vertex of subgraph S_j, where $i, j \leq n - k + 1$.*

Proof.

We also use induction to prove this lemma. Let A_y be the last vertex of subgraph S_1, B_y be the last vertex of subgraph S_2 and C_y be the last vertex of subgraph S_3 such that

$A_y = \{a_1, a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$,

$B_y = \{a_2, a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$, and

$C_y = \{a_3, a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$.

We shall get that the intersection between A_y and B_y is $\{a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$, the intersection between A_y and C_y is also $\{a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$, and the intersection between B_y and C_y is also $\{a_{(n-k+2)}, a_{(n-k+3)}, a_{(n-k+4)}, \ldots, a_n\}$ which has cardinality $k-1$. We can also say that among A_y, B_y, and C_y, they differ in only one element. For A_y, it differs one element from B_y because it does not have element a_2, and it differs one element from C_y because it does not have element a_3. For B_y, it differs one element from A_y because it does not have element a_1, and it differs one element from C_y because it does not have element a_3. Thus, A_y is adjacent to B_y and C_y. This adjacency also occurs between B_y and C_y, B_y and D_y, C_y and D_y, and so on, for D_y is the last vertex of subgraph S_4. Thus, the last vertex of subgraph S_i is always adjacent to the last vertex of subgraph S_j of the graph $G(n,k)$ of the Johnson Scheme, where $i, j \leq n-k+1$.

LEMMA 4. *The first vertex of subgraph $S_{(n-k)}$ is adjacent to the vertex of subgraph $S_{(n-k+1)}$.*

Proof.

Recall that in the last subgraph, $S_{(n-k+1)}$, has only one vertex. This vertex is the first vertex and also the last vertex of subgraph $S_{(n-k+1)}$. So, this lemma is already proven in **Lemma 3.2**.

THEOREM 1. *The graph $G(n,k)$ of the Johnson Scheme is hamiltonian.*

Proof.

We shall use the above lemmas to prove this theorem. Let S_i is the ith subgraph, for $i = 1, 2, \ldots, n-k+1$. If i is odd, we make an edge between the first vertex of subgraph S_i and the first vertex of subgraph $S_{(i+1)}$ (**Lemma 3.2**), If i is even, we make an edge between the last vertex of subgraph S_i and the last vertex of subgraph $S_{(i+1)}$ (**Lemma 3.3**). If i is equal to $n-k+1$, we make an edge between the last vertex of subgraph S_1 and the vertex of subgraph $S_{(n-k+1)}$. Now, every subgraph has been connected. At last, **Lemma 3.1** has proven that there is hamiltonian path in every subgraph. Thus, there is hamiltonian cycle in the graph $G(n,k)$ of the Johnson Scheme.

4. ONE-TO-ONE PERMUTATION ROUTING ON THE GRAPH $G(N,K)$ OF THE JOHNSON SCHEME

Recall these definitions:

The basic permutation routing problem is defined on a graph G as follows: for all i, vertex v_i wants to send a packet to vertex $d(v_i)$, where d is a permutation of the vertices of graph G.

One-to-one permutation routing of a graph G is defined as follows: every vertex initially sends at most one packet and is the final destination of at most one packet.

In our algorithm, we shall use the following:

- Store-and-forward routing model: In a graph, if a vertex has more than one packet which will go out using the same edge, then the packets can be stored at a vertex until they can use the edge.
- Bit-fixing routing strategy: In a graph, if a vertex v_i wants to send a packet p_i to a vertex $d(v_i)$, given that the vertex v_i's and vertex $d(v_i)$'s addresses are a set of numbers in ascending order, then scan the numbers of $d(v_i)$ from left to right, and compare them with the address of the current location of p_i. Send p_i out of the current vertex along the edge corresponding to the left-most numbers in which the current position and $d(v_i)$ differ.

The general idea of one-to-one permutation routing algorithm are:

1. We divide the graph $G(n,k)$ into subgraphs [18].
2. We route the packets within subgraphs simultaneously before sending to the destination vertex [8].

Thus, the one-to-one permutation routing algorithm for graph $G(n,k)$ of the Johnson Scheme is as follows:

Let source vertex be denoted by $v_i = \{a_1, a_2, \ldots, a_k\}$ and destination vertex be denoted by $d(v_i) = \{b_1, b_2, \ldots, b_k\}$.

1 Begin

2 If $a_1 \leq b_1$ then

3 Traverse from v_i to $w_i = \{a_1, b_2, \ldots, b_k\}$

along subgraph S_{a_1} using bit-fixing strategy;

4 If $a_1 \neq b_1$ then

5 Traverse from w_i to $d(v_i)$ in one step;

6 Else

7 Traverse from v_i to $y_i = \{b_1, a_2, \ldots, a_k\}$ in one step;

8 Traverse from y_i to $d(v_i)$ along subgraph S_{b_1} using

bit-fixing strategy;

9 End.

In this algorithm, we shall focus on step 3. In this step, we only focus on subgraph S_1, since in this subgraph we have the largest vertices, thus, will have the largest packets queue in a vertex. Step 8 will be handled similarly.

We shall determine the complexity of the routing scheme by getting the maximum number of packets accumulating and queuing at a vertex [8].

We shall consider this example with the big enough of n to show the number of packets queuing in a vertex.

LEMMA 5. *At most $(nk-k^2)(k-3)$ packets will queue in a vertex.*

Proof.

The packets which queue in a vertex x either will be sent to w_i or to other vertices through w_i. If they are sent to w_i, then they will be at most $(nk-k^2)$ packets, because there are $(n-k)$ of $d(v_i)$ which are adjacent to w_i in other subgraph and $(nk-k^2)-(n-k)$ which are adjacent to w_i in S_i. Otherwise, if they are sent to other vertices through w_i, then they will be at most $(nk-k^2)\times(k-4)$, because there are $(k-4)$ vertices will through the same edge of vertex w_i to receive packets and each of them will have $(nk-k^2)$ packets.

There are $(k-4)$ vertices will through the same edge of vertex w_i to receive packets because there are $(k-3)$ vertices v_i that send packets to vertex x at a time and queue.

There are $(k-3)$ vertices v_i that send packets to vertex x at the same time and queue, because for v_i to send packets to vertex x at the same time and queue, then a_1 and a_k in v_i and x should be the same element. And a_3 in v_i should also the same element as b_2 of x. And a_2 can be any numbers. At last, elements a_4 to $a_{(k-1)}$ in v_i are a combination of b_3 to $b_{(k-1)}$ of x. Assume without loss the generality that $a_1 < a_2 < a_3 < \cdots < a_k$ and $b_1 < b_2 < b_3 < \cdots < b_k$.

Thus, there are $(nk-k^2)\times(k-3)$ packets queueing in x.

This condition only occurs once, because there is no packet that will be sent to $d(v_i)$ (queue in vertex x) which will go through the same edge of vertex w_i.

Hence, we have this theorem.

THEOREM 2. *The one-to-one permutation routing on graph $G(n,k)$ of the Johnson Scheme can be done in $O(nk^2)$ time.*

Proof.

Lemma 3.5 clearly shows that at most $(nk-k^2)(k-3)$ packets queue in a vertex (this occurs on steps 3 and 8 of the algorithm). Since this worst case occurs once, thus, we need at most $(nk-k^2)(k-3) = O(nk^2)$ time to finish the one-to-one permutation routing on graph $G(n,k)$ of the Johnson Scheme.

5. GOSSIPING ON THE GRAPH $G(N,K)$ OF THE JOHNSON SCHEME

One-to-all broadcasting is an operation where a vertex (source) must send a packet to all other vertices of the graph. All-to-all broadcasting, also known as gossiping or total exchange, is a generalization of one-to-all broadcasting in which all vertices simultaneously initiate a broadcast. In other words, gossiping is the process whereby each vertex sends a packet to all the others.

Gossiping is done as follows:

1. *Initialization*: Let $G=(V,E)$ be a graph. With each vertex v, associate an initial packet.
2. *Allowable steps*: Each vertex can send its packet to a neighbor or neighbors and/or receive a packet from a neighbor or neighbors depending on the model of communication used. After receiving any packets, vertices take the union of all packets received at that step, thus forming new packets for the next step. At the other steps, when we take the union of all packets, we disregard multiplicities of packets.

The model of communication that we consider is *full-duplex* communication which means that a vertex can simultaneously send and receive on all its edges. In the way to determine the complexity of gossiping in a graph with this model of communication, the graph's diameter is the lower bound [21].

Intuitively, we can do gossip by this greedy algorithm:

```
1   For r = 1 to z Do
2     Every vertex gossip to all its adjacent vertices
```

This greedy algorithm will make the diameter of the graph $G(n,k)$ the tight bound, where $z=k$ if $k \le \lfloor\frac{n}{2}\rfloor$, and $z = n-k$ if $k > \lfloor\frac{n}{2}\rfloor$. However, this algorithm is not efficient, since at every step all vertices will broadcast in all its edges. In terms of interconnection network, we might not be able to do this all the time because of insufficient memory of processors or bandwidth.

We shall determine the number of exchanges in gossiping process to see the efficiency of an algorithm. In the greedy algorithm, the number of exchanges is $\Theta(n^{k+1}kd)$ where $d = min(k, n-k)$. This number of exchanges is obtained by each of the $\binom{n}{k}$ vertices sends to $nk-k^2$ vertices in d times. Thus, the total number of messages sent is $\binom{n}{k}(nk-k^2)d = \Theta(n^k)(nk-k^2)d = \Theta(n^{k+1}kd)$.

For reason of efficiency, we shall construct another algorithm to do gossiping on the graph $G(n,k)$.

Consider the clique on the graph $G(n,k)$ of the Johnson Scheme. Recall that a clique in a graph is a set of vertices where every pair is joined by an edge. Thus, every vertex in graph $G(n,k)$ which differ in one element obtain a clique. Then, vertices

$\{a_1,a_2,\dots,a_{(k-1)},a_k\},\{a_1,a_2,\dots,a_{(k-1)},a_{(k+1)}\},\dots,$
$\{a_1,a_2,\dots,a_{(k-1)},a_n\}$

form a clique, or vertices

$\{a_i,a_{(i+1)},\dots,a_{(k-2)},a_k,a_n\},\{a_i,a_{(i+1)},\dots,a_{(k-2)},a_k,$
$a_{(n-1)}\},\dots,\{a_i,a_{(i+1)},\dots,a_{(k-2)},a_k,a_{(k+1)}\}$

also form a clique. In general, a set of vertices that pairwise differ in one element form a clique on the graph $G(n,k)$.

There is a vertex in a clique that adjacent to a vertex in other clique, however, there is also no vertex in a clique that adjacent to any vertex in other clique. For example, consider *Figure 3.1*, We can say that {{1,2,3}, {1,2,4}, {1,2,5}, {1,2,6}, {1,2,7}} is a clique, say clique c_1. {{2,3,4}, {2,3,5}, {2,3,6}, {2,3,7}} is another clique, say clique c_2, and {{3,4,5}, {3,4,6}, {3,4,7}} is also a clique, say clique c_3. Clique c_1 has a vertex that adjacent to a vertex in clique c_2, but no vertex in c_1 that adjacent with any vertex in clique c_3.

Again, consider *Figure 3.1*, we make the whole cliques in graph $G(7,3)$ as follows:

Clique No.	Vertices
1	{1,2,3},{1,2,4},{1,2,5},{1,2,6},{1,2,7}
2	{1,3,7},{1,3,6},{1,3,5},{1,3,4}
3	{1,4,5},{1,4,6},{1,4,7}
4	{1,5,7},{1,5,6}
5	{2,3,4},{2,3,5},{2,3,6},{2,3,7}
6	{2,4,7},{2,4,6},{2,4,5}
7	{2,5,6},{2,5,7}
8	{3,4,5},{3,4,6},{3,4,7}
9	{3,5,7},{3,5,6}
10	{4,5,6},{4,5,7}
11	{1,6,7},{2,6,7},{3,6,7},{4,6,7},{5,6,7}

The last clique, clique number 11, c_{11}, we construct a clique from the last vertex of all subgraph S_i, which differ in the first element, since each of them has no partner. This will apply for any n and k of the graph $G(n,k)$.

Now, we construct the clique-gossiping algorithm on the graph $G(n,k)$.

```
1   For r = 1 to min(k, n − k) Do
2     Every vertex gossips within its clique;
3     Every vertex gossips with vertices of other cliques;
4   Every vertex gossips within its clique;
```

Consider these two cases.

Case 1: Assume $k \leq \lfloor \frac{n}{2} \rfloor$

LEMMA 6. *Assume $k \leq \lfloor \frac{n}{2} \rfloor$, then the graph G(n,k) will finish gossiping in $2k+1$ units time.*

Proof:

To prove this lemma, we use the clique-gossiping algorithm.

In step 2, every vertex will gossip within its clique. Since a clique is a complete graph where every vertex is an end-vertex for others, thus, this step will complete in 1 unit time. After this step, every vertex in a clique will know each other's packet.

Step 3, Every vertex gossips with vertices of other cliques. This step will complete also in 1 unit time. After this step, some vertices of a clique will know the packets of other cliques.

For example, vertex {1,6,7} of clique c_{11} will know the packets of cliques c_1, c_2, c_3, and c_4. However, it does not know the packets of other cliques. Meanwhile, vertex {2,6,7} will only know the packets of cliques c_5, c_6 and c_7 (means that we need to repeat step 1). Moreover, some vertices in clique c_1 will know the packet of cliques $c_1, c_2, c_3, c_4, c_5, c_6, c_7$ and c_{11}, but no vertex in c_1 knows the packets of clique c_8, c_9 and c_{10}, since there is no vertex in c_1 that adjacent with vertices in those clique. But clique c_{11}, which adjacent to c_1, knows the packets of those three clique (means that we need to repeat step 2).

We repeat steps 2 and 3 in k times because, in fact, the longest path for a clique to reach another clique is as many as the different element among their vertices. Since, there are cliques which their vertices different in k elements, thus, it needs k steps for a clique to reach another clique.

At last, we do step 4, which is the same as the second step, to gossip within a clique. We should do this step because after repeat k times, the last state of every clique is that every vertex in a clique will have different packets which it get when do the third step at time k. Thus, to complete the gossiping process every vertex should once again gossip within its clique.

Since we repeat steps 2 and 3 in k times, and each of the steps need 1 unit time, then steps 1 to 3 will take $2k$ units time. Plus 1 unit time of the forth step. Thus, totally we need $2k+1$ units time to finish gossiping.

Hence, we prove the lemma.

Case 2: Assume $k > \lfloor \frac{n}{2} \rfloor$

For this case, we will use the same algorithm.

But, we will repeat the process in $n-k$ times. Because in this case, $k > \lfloor \frac{n}{2} \rfloor$, the longest path for a clique to reach another clique is $n-k$.

So, in this case we also need $2(n-k)+1$ units time to finish gossiping.

Hence, we have this lemma.

LEMMA 7. *Assume $k > \lfloor \frac{n}{2} \rfloor$, then the graph G(n,k) will finish gossiping in $2(n-k)+1$ units time.*

We conclude **Lemma 3.6** and **Lemma 3.7** in this theorem.

THEOREM 3. *The gossiping process on the graph $G(n,k)$ will finish in $O(min(k, n-k))$ time.*

Now, consider the number of exchanges in the clique-gossiping algorithm.

In step 2, there will be $(n-k+1)^2$ message exchanges for every vertex to gossip within its clique, because in a clique there are at most $(n-k+1)$ vertices, and this occurs as many as d times, where $d = min(k, n-k)$.

In step 3, there are $(n-k+1)$ vertices in a clique, and each of them are connected to at most $\Theta(n^{k-3})$ vertices in other cliques. So, in this step, there will be $(n-k+1) \cdot \Theta(n^{k-3}) = \Theta(n^{k-2})$ message exchanges for every vertex of a clique to gossip with its adjacent vertices. This occurs as many as d times.

In step 4, there will be $(n-k+1)^2$ message exchanges for every vertex to gossip within its clique, because in a clique there are at most $(n-k+1)$ vertices, and this occurs only once.

Thus, the total number of exchanges is: $(n-k+1)^2 d + \Theta(n^{k-2})d + (n-k+1)^2 = \Theta(n^{k-2}d + dn^2)$. This means, the clique-gossiping algorithm is more efficient by the factor of $n^3 k$ than the greedy algorithm in terms of the number of exchanges.

The table below shows the result of the total number of exchanges of the clique-gossiping algorithm.

Instructions	No. of steps	No. of iterations
Every vertex gossips within its clique	$(n-k+1)^2$	d
Every vertex gossips with vertices of other cliques	$\Theta(n^{k-2})$	d
Every vertex gossips within its clique	$(n-k+1)^2$	1

Total number of exchanges = $\Theta(n^{k-2}d + dn^2)$.

Hence, we have this theorem.

THEOREM 4. *There is $\Theta(n^{k-2}d + dn^2)$ total number of exchanges in the clique-gossiping algorithm.*

6. SORTING ON THE GRAPH $G(N, K)$ OF THE JOHNSON SCHEME

Recall the definition of the sorting problem:

Input: A set of n packets, $p_1, p_2, \ldots, p_n$, with labels $l_1, l_2, \ldots, l_n$, for $l_i > 0$ and is integer. We write (p_1, l_1), means packet p_1 with label l_1.

Output: A permutation or reordering $(p'_1, l'_1), (p'_2, l'_2), \ldots, (p'_n, l'_n)$ of the input sequence such that $l'_1 \leq l'_2 \leq \cdots \leq l'_n$.

In terms of a graph G,

Input: Every vertex, $v_1, v_2, \ldots, v_n$, will have $(p_1, l_1), (p_2, l_2), \ldots, (p_n, l_n)$, respectively.

Output: Every vertex, $v_1, v_2, \ldots, v_n$, will have $(p'_1, l'_1), (p'_2, l'_2), \ldots, (p'_n, l'_n)$, respectively, where $l'_1 \leq l'_2 \leq \cdots \leq l'_n$.

We shall determine the order of the vertices as follows: In general, in every subgraph S_i, if i is odd then the order of the vertices is the same as the sequence of vertices in **Lemma 3.1**, however, if i is even then the order of the vertices is in the opposite order from the sequence of vertices in **Lemma 3.1**.

Our sorting algorithm is as follows:

We write (l_i, v_i) to denote that label l_i is in vertex v_i, for any i.

0 Initialization in every vertex: counter=0, vrecord=$\emptyset$, (p_{init}, l_{init}).

1 Do gossiping by sending (l_i, v_i) using the

clique-gossiping algorithm, during that time we do

2 If $l_{init} > l_i$ and $v_i \notin$ vrecord then

3 counter=counter+1;

4 vrecord=vrecord $\cup$ v_i;

{repeat steps 2 to 4 until all packets are compared};

5 Route (p_{init}, l_{init}) to $v_{(counter+1)}$ using the one-to-one

permutation routing algorithm;

6 If $v_{(counter+1)}$ receive more than one packet, say c, then

7 Route $c-1$ packets to $v_{(counter+2)}, \ldots, v_{(counter+c-1)}$ using

the one-to-one permutation routing algorithm;

First, we initialize in every vertex, counter=0, vrecord=$\emptyset$, and (p_{init}, l_{init}). Variable counter is used to count labels that less than l_{init}, thus, it will determine total packets less than its and also p_{init} destination vertex. Variable vrecord is used to store vertices number which pass a particular vertex and the label is counted. We write (p_{init}, l_{init}) to denote an initial packet and label in particular vertex, v_i.

Steps 1 to 4 are simultaneous and parallel processes. Every vertex gossips and compares its l_{init} with l_i and vrecord with v_i, if $l_i < l_{init}$ and $v_i \notin$ vrecord, then we increment the counter with 1, and store v_i in vrecord.

In step 5, we route every initial packet to its proper destination, in parallel, by using one-to-one permutation routing algorithm.

Steps 6 and 7, some vertices will route their packets, since they received more than one packet.

Now, we shall determine the complexity of the sorting algorithm by assuming the initial label is uniformly distributed.

In this assumption, the initial label is uniformly distributed over the interval [1,m], where $m = \binom{n}{k}$ and is integer. This means each number on the interval [1,m] is equally likely to occur. However, in fact, it is likely that a few numbers will appear more than once and a few will not appear at all [20].

During the gossiping process, in steps 2, we do the comparison process. However, since the gossiping process finish in $O(min(k, n-k))$ time and every vertex will have $\binom{n}{k}$ packets, then we still have $\binom{n}{k} - (2k+1)$ packets to be compared. Thus, we need $O(N)$ to determine the destination of p_i, where $N = \binom{n}{k}$.

As we mention before, we use the routing algorithm in step 5. It is not exactly one-to-one permutation routing, since there are a few packets will have the same destination (but it is very small (constant)). So, we need $O(nk^2)$ to finish routing.

Since it is only few packets will be received by a vertex, so, steps 6 and 7 will finish in $O(1)$ step.

Thus, the sorting algorithm takes $O(N) + O(nk^2) + O(1) = O(N)$ to finish sorting, where $N = \binom{n}{k}$.

Hence, we have this theorem.

THEOREM 5. *All packets in graph $G(n, k)$ of the Johnson Scheme will be sorted in $O(N)$ time, where $N = \binom{n}{k}$.*

As we know, the best parallel time possible for sorting in any graph with N vertices is $O(\log N)$, and the sequential time for sorting in any graph with N vertices is $O(N \log N)$.

7. REFERENCES

[1] Moon, A., *On the Uniqueness of the Graph $G(n, k)$ of the Johnson Scheme*, J. Combinatorial Theory, Series B, 1982.

[2] Chen, L. and Lih K.W., *Hamiltonian Uniform Subset Graph*, J. Combinatorial Theory, Series B, (1987).

[3] Zhang, F., Lin, G. and Cheng, R., *Some Distance Properties of the Graph $G(n, k)$ of Johnson Scheme*, presented in the International Congress of Algebra and Combinatorics 1997, Hongkong, 19-25 August 1997.

[4] Muga II, F.P., Caro, J.D.L., Adorna, H.N., and Baes, G., *On the Wide-Diameter of the Graph $G(n, k)$ of the Johnson Scheme of the First Order*, January 2000.

[5] Konig, J.C., Rao. P.S., Trystram, D., *Analysis of Gossiping Algorithms with Restricted Buffer*, Parallel Algorithm and Applications, Vol. 13, pp. 117-133, 1998.

[6] Monakhova, E.A., *Algorithm and Lower Bound for p-Gossiping in Circulant Networks*, SPAN 1997, pp.132-137, 1997.

[7] Leighton, T., *Theory of Parallel and VLSI Computation*, Lecture Notes, MIT, Sept. 1993.

[8] Wei, D.S.L., Muga II, F.P., Naik, K., *Isomorphism of Degree Four Cayley Graph and Wrapped Butterfly and Their Optimal Permutation Routing Algorithm*, IEEE Transactions on Parallel and Distributed Systems, Vol. 10, No. 11, pp. 1290-1298, Dec. 1999.

[9] Matwani, R., Raghavan, P., *Randomized Algorithms*, Cambridge University Press, 1995.

[10] Varma, A., Raghavendra, C.S., *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice*, IEEE Computer Society Press, 1994.

[11] Hartsfield, N., Ringel, G., *Pearls in Graph Theory*, Academic Press, 1994.

[12] Evans, J.R., Minieka, E., *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, Inc., 1992.

[13] Cameron, P.J., *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, 1994.

[14] Cormen, T.H., Leiserson, C.E., Rivest, R.L., *Introduction to Algorithms*, The MIT Press, 1990.

[15] Parthasarathy, K.R., *Basic Graph Theory*, Tata McGraw-Hill Publishing Company Limited, 1994.

[16] Chartrand, G., Oellermann, O.R., *Applied and Algorithmic Graph Theory*, McGraw-Hill, Inc., 1993.

[17] Krumme, D.W., Cybenko, G., Venkataraman, K.N., *Gossiping in Minimal Time*, SIAM J. Comput., Vol.21, No. 1, pp. 111-139, February 1992.

[18] Newman, I., Schuster, A., *Hot-Potato Algorithms for Permutation Routing*, IEEE Transactions on Parallel and Distributed Systems, Vol. 6, No. 11, November 1995.

[19] Vadapalli, P., Srimani, P.K., *Shortest Routing in Trivalent Graph Network*, Information Processing Letters 57, pp. 183-188, 1996.

[20] Sedgewick, R., *Algorithms*, Addison-Wesley Publishing Company, Inc., Second Edition, 1988.

[21] Bagchi, A., Schmeichel, E.F., Hakimi, S.L., *Gossiping with Multiple Sends and Receives*, Discrete Applied Mathematics 64, pp. 105-116, 1996.

[22] Bollobás, B., *Quo Vadis, Graph Theory?, The Future of Graph Theory*, Annals of Discrete Mathematics, 1993.

[23] Duato, J., Yalmanchili, S., Ni, L., *Interconnection Networks an Engineering Approach*, IEEE Computer Society Press, 1997.