金字塔網路於一個點或邊損壞時的泛迴路性質
Pancycles of the Pyramid Network with One Node or One Edge Fault

Yi-Ching Chen
National Chi Nan University
Department of Computer Science and Information Engineering
s0321511@ncnu.edu.tw

Dyi-Rong Duh
National Chi Nan University
Department of Computer Science and Information Engineering
drduh@ncnu.edu.tw

摘要

本篇論文在金字塔網路損壞一個點或邊時，提出一對演算法來改進先前的演算法，除了壞點是在一層金字塔網路的峯點外，其餘皆能建構出任意長度的迴路。換句話說，在某種限制下，金字塔網路於損壞一個點或邊的情形之下，仍然能夠嵌入所有可能長度的迴路。

關鍵詞：金字塔網路、互連網路、泛迴路、容錯性。

Abstract

In this paper, we derive a pair of algorithms to improve the algorithm proposed by Wu in 2001 such that cycles of all possible lengths in the pyramid network with one faulty node or edge can be constructed except the apex is the faulty node in the one-layer pyramid network. In other words, under some constraint, pyramid networks with one node or edge fault can still embed cycles of all possible lengths.

Keywords: Pyramid networks, interconnection networks, pancycles, fault tolerance.

1. Introduction

The topological structure of an interconnection network can be modeled by a graph [20]. The vertices and edges in graphs correspond to components and links in interconnection networks, respectively. Throughout this paper, we use network and graph, processor (node) and vertex, and link and edge, interchangeably. In the design and implementation of communication networks, parallel computing and VLSI, network topologies are always used to analysis the performance of networks.

Various performance measures can be used to evaluate a network topology such as degree, transmission delay, diameter, fault tolerance, routing function, embeddability, symmetry, extendability, and layout of VLSI, etc. Ideally, we want to design a symmetrical, regular, scalable and reliable network, which has lower degree, lower transmission delay, small diameter, easy routing function, and an efficient layout of VLSI, etc. Unfortunately, there are a lot of mutually conflicting requirements in designing the topology of an interconnection network. For example, a lower degree network usually has a larger diameter and longer transmission delay. Therefore, it is not easy to design an optimum topology with all good properties.

Pyramid networks, suggested by Dyer and Rosenfeld [9], are well-known networks in image processing, parallel and network computing [5], [8], [13], [15], [19]. Up to now, some pyramid networks have been built such as Cray T3D and T3E. In image processing [8], [13], both software structures and hardware architectures use the pyramid network. In parallel and network computing [5], [15], [19], there are many efficient algorithms that are developed in pyramid computers.

With some special mapping functions, some networks can be embedded to others [2], [6], [10], [11], [12], [14], [16]. Therefore, all algorithms developed in the former also can be used in the latter. In other words, the latter network can emulate the former network. The cycle and path are simplest networks often used in parallel computing and local area networks. Cycles and paths are suitable for the development of simple algorithms. Constructing cycles of lengths ranging from three to the size of a graph is the pancyclic problem. This problem of pyramid networks has been studied in [21].

Since node or link faults may happen when a network is put in use, it is very meaningfully to consider the faulty network. Topological properties of the faulty networks were investigated by literatures [2], [7], [11], [12], [14], [21]. These researches related to the faulty networks include computing diameter, fault diameter, and wide diameter, providing routing, multicasting, broadcasting, and embedding rules, and so on. We concentrate our attention on the fault-tolerant embedding cycles in pyramid networks.

The rest of this paper is organized as follows: In Section 2, some notations and definitions of this paper and some theorems about pancyclic properties of a graph G are given. The pyramid networks are introduced in Section 3. In Section 4, we first derive a pair of algorithms to construct cycles of lengths ranging from three to $5\times4^{k-1}$ in two consecutive layers k-1 and k of the n-layer pyramid network where $2\leq k\leq n$. Then, we combine the algorithms and the original method proposed in [21] to construct cycles of all possible lengths of the pyramid networks with one faulty node. Furthermore, in Section 5, we show that the pyramid networks with one faulty edge are still pancyclic. Finally, this paper is concluded in Section 6.

2. Preliminaries

In this section, we present those basic definitions in graphs that used in the paper. Also, some theorems about pancyclic properties of a graph G are given. We list all fundamental notations refer to [4], [20].

Definition 2.1. The ***vertex connectivity*** (***edge connectivity***) of a connected graph G is denoted by $\kappa(G)$ ($\kappa'(G)$) means that to make the graph G become disconnected at least $\kappa(G)$ vertices (($\kappa'(G)$) edges) should be removed.

Definition 2.2. A graph G is ***bipartite*** if its vertex set can be partitioned into two nonempty subsets X and Y such that each edge of G has one end in X and the other in Y. A ***complete bipartite graph*** is a bipartite graph such that two vertices are adjacent if and only if they are in different subsets. When the complete bipartite graph with $|X| = p$ and $|Y| = q$, we denote it by $K_{p,q}$.

Definition 2.3. A cycle that contains every vertex of G exactly once is called ***Hamiltonian cycle***. A graph is ***Hamiltonian*** if it contains a Hamiltonian cycle.

Definition 2.4. A graph G is ***pancyclic*** if it contains cycles of lengths l for $3\leq l\leq|V(G)|$.

In the following, we list some theorems about pancyclic properties of a graph G.

Theorem 2.1 [3]. If G is a Hamiltonian graph of order p and size $q \geq p^2/4$, then either G is pancyclic or p is even and G is $K_{p/2, p/2}$.

Let ***deg(v)*** denote the ***degree*** of a vertex v in graph G, another two theorems related to pan cyclic problem are state below.

Theorem 2.2 [17]. Let G be a graph of order $n\geq3$ with a Hamiltonian cycle $(v_1, v_2, \ldots, v_n, v_1)$ where v_i.

(1) If $\deg(v_1)+\deg(v_n) \geq n$, then G is either pancyclic, bipartite, or missing only an $(n-1)$-cycle.
(2) If $\deg(v_1)+\deg(v_n) > n$, then G is pancyclic.

Theorem 2.3 [22]. Let G be a graph of order n containing a Hamiltonian cycle C. If x, y, and z are three consecutive vertices on C with $\deg(x)+\deg(z) > n$, then G is pancyclic.

3. Pyramid Networks

The pyramid network is a hierarchy structure based on mesh networks. So before introducing the pyramid network, we first describe the structure of the mesh network. The mesh network is defined as the Cartesian product $P_m\times P_n$ [20], denoted by $\boldsymbol{M(m, n)}$, where P_m and P_n are undirected paths with m and n vertices, respectively. The vertex set of mesh is $V(M(m, n)) = \{(x, y) \mid 1\leq x\leq m,\ 1\leq y\leq n\}$. And the edge set of mesh is $E(M(m, n)) = \{(x_1, y_1)(x_2, y_2) \mid |x_1-x_2| + |y_1-y_2| =1\}$. The maximum degree and minimum degree of a mesh are four and two, respectively.

We denote the n-layer pyramid network by ***PM*[*n*]**. Every ith layer of $PM[n]$ is an $M(2^i, 2^i)$, for $0\leq i\leq n$. The vertex set of $PM[n]$ is $V(PM[n]) = \{(k; x, y) \mid 0\leq k\leq n,\ 1\leq x\leq 2^k,\ 1\leq y\leq 2^k\}$. The edge $(k_1; x_1, y_1)(k_2; x_2, y_2)$ of $PM[n]$ satisfy one of the following statements:

(1) $k_1 = k_2$, $|x_1 - x_2| + |y_1 - y_2| =1$,
(2) $k_2 = k_1 + 1$, $x_1 = \lfloor x_2/2\rfloor$, $y_1 = \lfloor y_2/2\rfloor$.

We call the edge satisfies (1) a ***mesh-link*** and satisfies (2) a ***layer-link***. Let $(k_1; x_1, y_1)$ and $(k_1+1; x_2, y_2)$ be the two end nodes of a layer-link in $PM[n]$. Then, $(k_1; x_1, y_1)$ is the ***parent*** of $(k_1+1; x_2, y_2)$. An ***internal mesh-link*** uv represents that nodes u and v have a common parent. Otherwise, the rest of the mesh-links are ***external mesh-links***.

Furthermore, $PM[n]$ can be considered as a

4-ary rooted tree and every node in the same layer are connected as a mesh [1]. Thus, we have following properties.

(1) The ***apex*** (0; 1, 1) is connected to its four children. The degree of apex is four.

(2) Every node at a layer 1 to layer n–1 is connected to one link with the parent, at most four links with its siblings, and four links with its children. In this case, the maximum degree and minimum degree are nine and seven, respectively.

(3) Every node at layer n is connected to one link with its parent and at most four links with its siblings. In this case, the maximum degree and minimum degree are five and three, respectively.

(4) $|V(PM[n])| = \sum_{i=0}^{n} 4^i$ and $|E(PM[n])| = 4^{n+1}-2^{n+2}$ [21].

With all of theorems described in Section 2, we cannot examine whether the pyramid network is a pancyclic graph or not. [18] and [21] have shown that the pyramid network is a Hamiltonian graph by constructing a spanning cycle in it. In [21], Wu also show that $PM[n]$ is Hamiltonian-connected and pancyclic. By [7], we know that the vertex connectivity and edge connectivity of $PM[n]$ is three. And the vertex connectivity and edge connectivity of cycles equal to two. Therefore, only one node or one edge could be failure or the faulty pyramid network is not a pancyclic network. We will show that $PM[n]$ with one node or one edge fault is almost pancyclic in Section 4 and Section 5.

4. Pancycles of Pyramid Networks with One Node Fault

In the following, any two consecutive layers k and k–1 of a pyramid network is denoted by $\boldsymbol{PM[k;\, k-1]}$, where $k \geq 2$. In Subsection 4.1, we propose a pair of algorithms to construct all cycles in $PM[k; k-1]$ whose lengths are from three to the number of its nodes. In Subsection 4.2, we show that $PM[k; k-1]$ with a faulty node is pancyclic. In Subsection 4.3, we construct all cycles whose lengths are at most $|V(PM[n])|-1$ in $PM[n]$ with a faulty node.

4.1 Pancycles of $PM[k; k-1]$

We derive a pair of algorithms to construct all cycles in $PM[k; k-1]$ whose lengths are from three to the number of its nodes. We name the pair of algorithms PLPM1 and PLPM2 since they can construct *Pancycles in two consecutive Layers of PM[n]*. Both PLPM1 and PLPM2 input a layer k and a cycle length l, and then can return a cycle C_l whose length is l for $3 \leq l \leq 4^k + 4^{k-1} = 5\times 4^{k-1}$, where $2 \leq k \leq n$. Note that in the rest of this subsection we use symbol $\cup$ to denote union operation of ordering sets.

Algorithm PLPM1(*k*, *l*){
Initialization:
Three ordering sets $U_1(x, y) = \{(k; 2x-1, 2y-1), (k; 2x-1, 2y), (k; 2x, 2y), (k; 2x, 2y-1)\}$, $U_2(x, y) = \{(k; 2x, 2y), (k; 2x, 2y-1), (k; 2x-1, 2y-1), (k; 2x-1, 2y)\}$, and $U_3(x, y) = \{(k; 2x-1, 2y-1), (k; 2x, 2y-1), (k; 2x, 2y), (k; 2x-1, 2y)\}$.
Two ordering sets $U = U_1(1, 1)$ and $V = \{(k-1; 1, 1)\}$.
1. $l' = \text{Mul5}(l)$;
2. Let the first element in V be $(k-1; x, y)$.
FOR ($|U|+|V| < l'$){
2.1 IF (y is odd){
IF ($x < 2^{k-1}-1$) $x = x + 1$; $U = U \cup U_1(x, y)$;
IF ($x = 2^{k-1}-1$) $x = x + 1$; $U = U \cup U_3(x, y)$;
IF ($x = 2^{k-1}$) $y = y + 1$; $U = U \cup U_3(x, y)$;
}// End of IF (y is odd)
2.2 IF (y is even){
IF ($x = 1$) $y = y + 1$; $U = U \cup U_1(x, y)$;
IF ($1<x\leq 2^{k-1}$) $x = x - 1$; $U = U \cup U_2(x, y)$;
}// End of IF (y is even)
2.3 $V = (k-1; x, y) \cup V$;
}//End of FOR ($|U|+|V| < l'$).
3. $C_l = \text{Output}(U, V, l)$;
}//End of PLPM1

Algorithm PLPM2(*k*, *l*){
Initialization:
Three ordering sets $U'_1(x, y) = \{(k; 2x-1, 2y), (k; 2x-1, 2y-1), (k; 2x, 2y-1), (k; 2x, 2y)\}$, $U'_2(x, y) = \{(k; 2x, 2y-1), (k; 2x, 2y), (k; 2x-1, 2y), (k; 2x-1, 2y-1)\}$, and $U'_3(x, y) = \{(k; 2x, 2y-1), (k; 2x-1, 2y-1), (k; 2x-1, 2y), (k; 2x, 2y)\}$.
Two ordering sets $U = U'_3(1, 1)$ and $V = \{(k-1; 1, 1)\}$.
1. $l' = \text{Mul5}(l)$;
2. Let the first element in V be $(k-1; x, y)$.
FOR ($|U|+|V| < l'$){
2.1 IF (y is odd){
IF ($x < 2^{k-1}$) $x = x + 1$; $U = U \cup U'_1(x, y)$;
IF ($x = 2^{k-1}$) $y = y + 1$; $U = U \cup U'_2(x, y)$;
}//End of IF (y is odd)
2.2 IF (y is even){
IF ($x = 1$) $y = y + 1$; $U = U \cup U'_3(x, y)$;
IF ($x = 2$) $x = x - 1$; $U = U \cup U'_3(x, y)$;
IF ($2<x\leq 2^{k-1}$) $x = x - 1$; $U = U \cup U'_2(x, y)$;
}//End of IF (y is even)
2.3 $V = (k-1; x, y) \cup V$;
}//End of FOR ($|U|+|V| < l'$)
3. $C_l = \text{Output}(U, V, l)$;
}//End of PLPM2

Function Mul5(*l*){//normalize l to $l' = 5k$, k is a

positive integer.
SWITCH(l mod 5){
  Case 0: return l;
  Case 1: return $l + 4$;
  Case 2: return $l + 3$;
  Case 3: return $l + 2$;
  Case 4: return $l + 1$;
  }//End of SWITCH(l mod 5)
}//**End of Procedure Mul5(l)**

Procedure Output(U, V, l){
SWITCH(l mod 5){
  Case 0: Do nothing.
  Case 1: Delete the first and last three elements of U.
  Case 2: Delete the last three elements of U.
  Case 3: Delete the last two elements of U.
  Case 4: Delete the last element of U.
  }//End of SWITCH(l mod 5)
Output all elements of U and V orderly.
}//**End of Procedure Output(U, V, l)**

In PLPM1 and PLPM2, two ordering sets U and V are used to store all nodes whose elements are needed to form a cycle with length l'. Initially, U and V have four elements and one element, respectively. First, we use Function Mul5 to modify l to l' such that l' is a multiple of 5. Then, in each iteration, four elements are inserted into the rear of U and one element are inserted into the front of V. Note that the elements of U and V are in layer k and layer k–1 of $PM[n]$, respectively. The loop doesn't stop until $|U|+|V| = l'$. Finally, Procedure Output is used to output those elements in U and V that are necessary to form C_l.

By PLPM1 or PLPM2, we can easily obtain cycles of lengths ranging from 3 to the number of nodes in the two consecutive layers of pyramid networks. We have the following lemma.

Lemma 4.1. $PM[k; k–1]$, $k≥2$, of $PM[n]$ is a pancyclic network.

4.2 Pancycles of $PM[k; k–1]$ with a Faulty Node

In Subsection 4.1, we have already developed a pair of algorithms PLPM1 and PLPM2 to construct all cycles of lengths ranging from 3 to $5×4^{k–1}$ in $PM[k; k–1]$. These two algorithms are very similar except the rules they add the nodes that C_l needs. For example, PLPM1 appends $U_1(x, y)$ and PLPM2 appends $U'_1(x, y)$ to the rear of U in the same condition. In this subsection, we'd like to show that the faulty node at layer k of $PM[k; k–1]$ could be avoided by combining PLPM1 with PLPM2. In order to avoid the faulty node, we propose a new algorithm to construct *Pancycles on two consecutive Layers k and k+1 of PM[n] with one Faulty node at layer k (PLPMF for short)*. The proposed algorithm PLPMF is also shown how to combine PLPM1 with PLPM2 to establish a cycle in $PM[k; k–1]$.

As shown in [21], the coordinate of every node in $PM[n]$ can be clockwise rotated with 90°, 180°, or 270°. If we clockwise rotate the node $(k; x, y)$ with 90°, 180°, or 270°, then $(k; x, y)$ becomes $(k; 2^k–y+1, x)$, $(k; 2^k–x+1, 2^k–y+1)$, or $(k; y, 2^k–x+1)$, respectively. Conversely, the node $(k; x, y)$ can stand for nodes $(k; 2^k–y+1, x)$, $(k; 2^k–x+1, 2^k–y+1)$, and $(k; y, 2^k–x+1)$. Let $f=(k; x, y)$ be the faulty node and $p=(k–1; u, v)$ its parent where $1≤x, y≤2^k$ and $1≤u, v≤2^{k–1}$. Then five cases should be discussed according to the coordinate of f. There are four cases for $1 ≤ x, y ≤ 2$ and one case for $3 ≤ x ≤ 2^k–2$ and $1 ≤ y ≤ 2^k$. We do not care the nodes of $1 ≤ x ≤ 2$ and $1 ≤ y ≤ 2^k$ or the nodes of $2^k–1 ≤ x ≤2^k$ and $1 ≤ y ≤ 2^k$ since they can be mapped to the nodes that must be cared. Algorithm PLPMF is given below.

Algorithm PLPMF(k, f, l){
/* Input: a layer k where $2≤k≤n$, a faulty node $f = (k; x, y)$, and a length l of a cycle.
  Output: a cycle C_l. */
1. Apply PLPM1(k, l) but the elements of U and V are not outputted. If $f \notin U$ then output all elements of U and then all elements of V, and exit the procedure.
2. Apply PLPM2(k, l) but the elements of U and V are not outputted. If $f \notin U$ then output all elements of U and then all elements of V, and exit the procedure.
3. Case 1. (x=1 and y=1): Apply PLPM1(k, l+1) but the elements of U and V are not outputted. Remove f from U and then output all elements of U and then all elements of V.
   Case 2. (x=2 and y=2): Apply PLPM1(k, l+1) but the elements of U and V are not outputted. Let $U=U–U_1(1, 1)$ and then let $U=\{(k; 1, 1), (k; 2, 1)\}\cup U$. If l mod 5 $\neq$ 0 then let $U=\{(k; 1, 2)\}\cup U$. Output all elements of U and then all elements of V.
   Case 3. (x=2 and y=1): Apply PLPM2(k, l+1) but the elements of U and V are not outputted. Remove f from U and then output all elements of U and then all elements of V.
   Case 4. (x=1 and y=2): Apply PLPM2(k, l+1) but the elements of U and V are not outputted. Let $U=U–U'_3(1, 1)$ and then let $U=\{(k; 2, 1), (k; 2, 2)\}\cup U$. IF l mod 5 $\neq$ 0 then let $U=\{(k; 1, 1)\}\cup U$. Output all elements of U and then all elements of V.
   Case 5. ($3≤x≤2^k–2$ and $1≤y≤2^k$): Apply PLPM1(k, $5×4^{k–1}$) but the elements of U and

V are not outputted and let $U'=U$ and $V'=V$. Apply PLPM2(k, $5\times4^{k-1}$) but the elements of U and then all elements of V are not outputted.

Subcase 5.1. (x=2u–1 and y=2v–1, or x=2u and y=2v): Remove f and all elements after f from U and remove all elements before p from V. Remove f and all elements before f from U' and remove p and all elements after p from V'. Let $U=U\cup U'$ and $V=V\cup V'$. Output all elements of U and then all elements of V.

Subcase 5.2. (x=2u and y=2v–1, or x=2u–1 and y=2v): Remove f and all elements after f from U' and remove p and all elements before p from V'. Remove f and all elements before f from U and remove all elements after p from V. Let $U=U'\cup U$ and $V=V'\cup V$. Output all elements of U and then all elements of V.

}**//End of PLPMF**

The first two steps of Algorithm PLPMF are used to form C_ls that do not contain f in $PM[k; k–1]$ without any faulty node. In the third step of Algorithm PLPMF, five cases are discussed. Cases 1 and 3 apply PLPM1 and PLPM2 with length l+1, respectively. Since f is the first element of U. Therefore, after removing f from U, the cycle of length l can be easily established. In Case 2 (Case 4), after applying PLPM1 (PLPM2) with length l+1, we first remove four nodes from U and then insert two nodes into the front of U for reconstructing the cycle to avoid the faulty node f. If the length l is not a multiple of 5, one more node has to be inserted into the front of U for matching $|U|+|V| = l$. Case 5 is divided into two subcases and both of them apply PLPM1 and PLPM2 with $l=5\times4^{k-1}$ first. Two ordering sets U' and V' (U and V) are generated by PLPM1 (PLPM2). Base on the coordinates of the faulty node f and its parent p, we delete all nodes in U', V', U, and V that are not necessary to construct the desired cycle. After combining U (V) with U' (V'), C_l can be formed by outputting the all elements of U and V. With the add of PLPMF, we can constructed all cycles of lengths range from three to $5\times4^{k-1}–1$ in $PM[k; k–1]$ with one faulty node $f = (k; x, y)$. By Algorithm PLPMF, we have the following Lemma.

Lemma 4.2. $PM[k; k–1]$ of $PM[n]$, where $2\le k\le n$, with one faulty node is pancyclic.

4.3 Pancycles of $PM[n]$ with One Faulty Node

For convenience, let $PM_f[n]$ or $PM_{(k;\,x,\,y)}[n]$ denote $PM[n]$ with one faulty node $f=(k; x, y)$. In this subsection, we will show $PM_{(k;\,x,\,y)}[n]$ is pancyclic, where $0 \le k \le n$, except $PM_{(1;\,0,\,0)}[1]$. Due to $PM_{(1;\,0,\,0)}[1]$ does not contain C_4. Thus $PM_{(1;\,0,\,0)}[1]$ is not pancyclic. Obviously, $PM[1]$ with $f \ne (1; 0, 0)$ contains C_3 and C_4 , and is pancyclic. Therefore, we only need to construct all cycles of lengths ranging from three to $|V(PM(n))|–1$ in $PM_f[n]$ for $n \ge 2$. We first provide an algorithm to construct C_l in $PM_{(k;\,x,\,y)}[n]$ for $k \ge 3$ and $3 \le l \le |V(PM(n))|–1$. Since the algorithm can construct *Pancycles in PM[n] with one node Fault*, so it can be named *PPMF*. For $0 \le k \le 2 \le n$, we also propose an algorithm PPMF1 to construct C_l in $PM_{(k;\,x,\,y)}[n]$ where $3\le l\le|V(PM(n))|–1$.

By the result in Subsection 4.2, cycles of lengths ranging from 3 to $5\times4^{k-1}–1$ can be constructed in $PM[k; k–1]$ with one faulty node. Wu's algorithm can construct all cycles of lengths ranging from 3 to $|V(PM(n))|$ in $PM[n]$ without fault [21]. According to lengths of cycles, four cases are discussed in Algorithm PPMF. Algorithm PPMF is now given.

Algorithm PPMF(l, f, n){// $PM_{(k;\,x,\,y)}[n]$ for $k \ge 3$

/*Input: The cycle length l, where $3 \le l \le |V(PM[n])|–1$, the faulty node $f = (k; x, y)$, the layer n, $3 \le k \le n$.

Output: The cycle C_l. */

Case 1. ($3 \le l \le \sum_{i=0}^{k-1} 4^i = \frac{4^k–1}{3}$): Apply Wu's algorithm to form C_l.

Case 2. ($\sum_{i=0}^{k-1} 4^i < l \le \sum_{i=0}^{k} 4^i–1$): Construct a Hamiltonian cycle of length $\sum_{i=0}^{k-2} 4^i$ in $PM[k–2]$ by applying Wu's algorithm. Apply PLPMF(k, f, $l–\sum_{i=0}^{k-2} 4^i$) to construct a cycle of length $l–\sum_{i=0}^{k-2} 4^i$. Merge these two cycles to form C_l by removing edges (k–1; 2, 1)(k–1; 3, 1) and (k–2; 1, 1)(k–2; 2, 1), and then adding edges (k–2; 1, 1)(k–1; 2, 1) and (k–2; 2, 1)(k–1; 3, 1).

Case 3. ($l = \sum_{i=0}^{k} 4^i$): Construct a cycle of length $\sum_{i=0}^{k} 4^i–1$ by the steps of Case 2. Remove the apex (0; 1, 1) from the cycle by removing edges (0; 1, 1)(1; 1, 2) and (0; 1, 1)(1; 2, 1), and then adding the edge (1; 1, 2)(1; 2, 1). Construct a path with 2 nodes between (k+1; 2^{k+1}, 2) and (k+1; 2^{k+1}, 3) by applying Wu's algorithm. Merge the cycle and the path to form C_l by removing the edge (k; 2^k, 1)(k; 2^k, 2), and then adding edges (k; 2^k, 1)(k+1; 2^{k+1}, 2) and (k; 2^k, 2)(k+1; 2^{k+1}, 3).

Case 4. ($\sum_{i=0}^{k} 4^i < l \le 4^n$): Construct a cycle of length $\sum_{i=0}^{k} 4^i-1$ by the steps of Case 2. Construct a path with $l-\sum_{i=0}^{k} 4^i+1$ nodes between nodes $(k+1; 2^{k+1}, 1)$ and $(k+1; 2^{k+1}, 3)$ ($(k+1; 2^{k+1}, 2)$ and $(k+1; 2^{k+1}, 3)$) by applying Wu's algorithm for $l-\sum_{i=0}^{k} 4^i+1$ is odd (even). Merge the cycle and the path to form C_l by removing the edge $(k; 2^k, 1)(k; 2^k, 2)$, and then adding edges $(k; 2^k, 1)(k+1; 2^{k+1}, 1)$ and $(k; 2^k, 2)(k+1; 2^{k+1}, 3)$ or $(k; 2^k, 1)(k+1; 2^{k+1}, 2)$ and $(k; 2^k, 2)(k+1; 2^{k+1}, 3)$ for $l-\sum_{i=0}^{k} 4^i+1$ is odd (even).

}**//End of PPMF**

In Algorithm PPMF, the cycle of length $l<\sum_{i=0}^{k-1} 4^i$ can be constructed by applying Wu's algorithm as shown in Case 1. In Case 2, we construct two cycles by applying two different algorithms. Since $l-\sum_{i=0}^{k-2} 4^i > 15$ and the layer k of $PM_{(k; x, y)}[n]$ contains the edge $(k-1; 2, 1)(k-1; 3, 1)$ for $k \geq 3$, so these two cycles can be merged to form C_l. Case 3 and Case 4 are very similar except Case 3 need to remove the apex such that C_l can be constructed. Also the cycles constructed by PLPM1 and PLPM2 contain the edge $(k; 2^k, 1)(k; 2^k, 2)$. Therefore, the constructed cycles and path can be merged to form C_l.

Next, we explain how to construct C_l in $PM_{(k; x, y)}[n]$ where $3 \leq l \leq |V(PM(n))|-1$ for $0 \leq k \leq 2 \leq n$. Base on the value of k, three cases are considered in Algorithm PPMF1. Due to the node $(k; x, y)$ can represent nodes $(k; 2^k-y+1, x)$, $(k; 2^k-x+1, 2^k-y+1)$, and $(k; y, 2^k-x+1)$. Without loss of generality, assume that the faulty node f is $(1; 1, 1)$ for $k=1$ and f is $(2; x, y)$, $1 \leq x, y \leq 2$, for $k=2$. We give Algorithm PPMF1 as follows.

Algorithm PPMF1(*l, f, n*){// $PM_{(k; x, y)}[n]$ for $0 \leq k \leq 2 \leq n$

/*Input: The cycle length l, where $3 \leq l \leq |V(PM[n])|-1$, the faulty node $f = (k; x, y)$, the layer n, $2 \leq k \leq n$.

Output: The cycle C_l. */

Case 1. ($k = 0$): Construct a cycle of length l ($\sum_{i=1}^{2} 4^i$) in $PM_{(0; 1, 1)}[2]$ by applying PLPM1 for $l \leq \sum_{i=1}^{2} 4^i$ ($l > \sum_{i=1}^{2} 4^i$). If l is at most $\sum_{i=1}^{2} 4^i$ then let C_l be the constructed cycle and exit the procedure. If $l=\sum_{i=1}^{2} 4^i+1$ then let $l = l+1$ and delete the node $(2; 1, 1)$ from C_l by removing edges $(1; 1, 1)(2; 1, 1)$ and $(2; 1, 1)(2; 1, 2)$, and adding the edge $(1; 1, 1)(2; 1, 2)$. Construct a path with $l-\sum_{i=1}^{2} 4^i$ nodes between $(3; 2^3, 1)$ and $(3; 2^3, 3)$ ($(3; 2^3, 2)$ and $(3; 2^3, 3)$) by applying Wu's algorithm for $l-\sum_{i=1}^{2} 4^i$ is odd (even). Merge the cycle and the path to form the desired cycle by removing the edge $(2; 2^2, 1)(2; 2^2, 2)$, and then adding edges $(2; 2^2, 1)(3; 2^3, 1)$ and $(2; 2^2, 2)(3; 2^3, 3)$ ($(2; 2^2, 1)(3; 2^3, 2)$ and $(2; 2^2, 2)(3; 2^3, 3)$) for $l-\sum_{i=1}^{2} 4^i$ is odd (even).

Case 2. ($k = 1$): Construct C_3 (C_4) which contains the node $(1; 2, 1)$ in $PM_{(1; 1, 1)}[1]$ for l is odd (even). If l is at most 4 then let C_l be the constructed cycle and exit the procedure. If $l=5$ then let $l = l+1$ and delete the node $(1; 1, 2)$ from C_4 by removing edges $(0; 1, 1)(1; 1, 2)$ and $(1; 1, 2)(1; 2, 2)$, and adding the edge $(0; 1, 1)(2; 1, 2)$. Construct a path with $l-4$ nodes between $(2; 2^2, 1)$ and $(2; 2^2, 3)$ ($(2; 2^2, 2)$ and $(2; 2^2, 3)$) by applying Wu's algorithm for $l-4$ is odd (even). Merge the cycle and the path to form the desired cycle by removing the edge $(1; 2, 1)(1; 2, 2)$, and then adding edges $(1; 2, 1)(2; 2^2, 1)$ and $(1; 2, 2)(2; 2^2, 3)$ ($(1; 2, 1)(2; 2^2, 2)$ and $(1; 2, 2)(2; 2^2, 3)$) for $l-4$ is odd (even).

Case 3. ($k = 2$): Construct a cycle of length $l-1$ ($\sum_{i=1}^{2} 4^i-1$) in $PM[2; 1]$ with $f=(0; 1, 1)$ by applying PLPMF for $l \leq \sum_{i=1}^{2} 4^i$ ($l > \sum_{i=1}^{2} 4^i$). If $l \neq \sum_{i=1}^{2} 4^i+1$ then add the apex $(0; 1, 1)$ into the cycle by removing the edge $(1; 1, 1)(1; 1, 2)$ and then adding edges $(0; 1, 1)(1; 1, 1)$ and $(0; 1, 1)(1; 1, 2)$. If l is at most $\sum_{i=1}^{2} 4^i$ then let C_l be the constructed cycle and exit the procedure. If $l=\sum_{i=1}^{2} 4^i+1$ then let $l = l+1$. Construct a path with $l-\sum_{i=1}^{2} 4^i$ nodes between $(3; 2^3, 1)$ and $(3; 2^3, 3)$ ($(3; 2^3, 2)$ and $(3; 2^3, 3)$) by applying Wu's algorithm for $l-\sum_{i=1}^{2} 4^i$ is odd (even). Merge the cycle and the path to form the desired cycle by removing the edge $(2; 2^2, 1)(2; 2^2, 2)$, and then adding edges $(2; 2^2, 1)(3; 2^3, 1)$ and $(2; 2^2, 2)(3; 2^3, 3)$ ($(2; 2^2, 1)(3; 2^3, 2)$ and $(2; 2^2, 2)(3; 2^3, 3)$) for $l-\sum_{i=1}^{2} 4^i$ is odd (even).

}**//End of PPMF1**

Note that we cannot merge a node in layer $k+1$ of $PM_{(k; x, y)}[n]$ with a cycle whose nodes are all in layer at most k. Thus we have to take care

l=5 ($l=\sum_{i=1}^{2} 4^i+1$) for k=1 (k=0, 2) in Algorithm PPMF1. With the aid of Algorithm PPMF and Algorithm PPMF1, we know that $PM_{(k;\ x,\ y)}[n]$ contains all cycles of lengths ranging from 3 to $|V(PM[n])|-1$ where $n \geq 2$. According to the discussion above, we have the following theorem.

Theorem 4.1. $PM_{(k;\ x,\ y)}[n]$ is pancyclic except $PM_{(1;\ 0,\ 0)}[1]$ where $0 \leq k \leq n$.

5. Pancycles of the Pyramid Networks with One Edge Fault

In this section, we use the result in Subsection 4.2 to show how to construct all cycles of lengths ranging from three to $|V(PM[n])|$ in $PM[n]$ with one edge fault. Let $PM_e[n]$ or $PM_{(k;\ x_1,\ y_1)(k;\ x_2,\ y_2)}[n]$ denote $PM[n]$ with a faulty edge $e = (k; x_1, y_1)\ (k; x_2, y_2)$. And now we discuss the pancyclic problem of $PM_e[n]$. As described in Section 3, all links in pyramid networks are divided into three parts: external mesh-links, internal mesh-links, and layer-links. External mesh-links are used in [21] to construct all cycles ranging from three to $|V(PM(n))|$ are called ***used external mesh-link.*** Clearly, if the faulty edge $e = (k; x_1, y_1)\ (k; x_2, y_2)$ is an external mesh-link where $2 \leq k \leq n$, we can mark them as follows.

1. Column ($x_1 = x_2$): $(k; 2^k-1, 1)(k; 2^k-1, 2)$, $(k; 2^k, 1)(k; 2^k, 2)$, $(k; 1, 4i)(k; 1, 4i+1)$, $(k; 2, 4i)(k; 2, 4i+1)$, and $(k; 2^k-1, 4i+2)(k; 2^k-1, 4i+3)$, $(k; 2^k, 4i+2)(k; 2^k, 4i+3)$ for $i = 1..k-2$. Note that only two external mesh-links (2; 3, 1)(2; 3, 2) and (2; 4, 1)(2; 4, 2) in columns are used if $k = 2$.
2. Row ($y_1 = y_2$): All external mesh-links in rows are used but $(k; 2, 1)(k; 3, 1)$. Note that the edge $(k; 1, 1)(k; 1, 2)$ may be or may not be used depending on the length of the desired cycle.

The used external mesh-links in layer 3 of $PM[n]$ are marked as follows:

1. Column edges: (7, 2)(7, 3), (8, 2)(8, 3), (1, 4)(1, 5), (2, 4)(2, 5), (7, 6)(7, 7), (8, 6)(8, 7).
2. Row edges: All external mesh-links in rows are used but (3; 2, 1)(3; 3, 1). The gray line is the edge (3; 1, 1)(3; 1, 2).

Since the node $(k; x, y)$ of $PM[n]$ can represent nodes $(k; 2^k-y+1, x)$, $(k; 2^k-x+1, 2^k-y+1)$, and $(k; y, 2^k-x+1)$ by clockwise rotating it with 90°, 180°, and 270°, respectively. Therefore, we can define a mapping function τ to map a used external mesh-link, which is the faulty edge, into an unused external mesh-link. Without loss of generality, we assume that the used external mesh-link is $e = (k; x_1, y_1)(k; x_2, y_2)$ in layer k of $PM_e[n]$. The mapping function τ is defined according to the location of e as follows:

Case 1. (e is in a column): $\tau((k; x_1, y_1)(k; x_2, y_2)) = (k; 2^k-x_1+1, 2^k-y_1+1)(k; 2^k-x_2+1, 2^k-y_2+1)$

Case 2. (e is in a row and $x_1 = 4i-2$, $x_2 = 4i-1$ for i=1..k–1): $\tau((k; x_1, y_1)(k; x_2, y_2)) = (k; y_1, 2^k-x_1+1)(k; y_2, 2^k-x_2+1)$

Case 3. (e is in a row and $x_1 = 4i$, $x_2 = 4i+1$ for i=1..k–2): $\tau((k; x_1, y_1)(k; x_2, y_2)) = (k; 2^k-y_1+1, x_1)(k; 2^k-y_2+1, x_2)$

In Case 1, the faulty edge e is in a column and can be mapped to an unused edge by clockwise rotating it with 180°. Case 2 and Case 3 are concerned about the faulty edge e in a row. If x_1 is not a multiple of 4 the mapping function do the clockwise rotation with 270° in Case 2. Otherwise, x_1 is a multiple of 4; the faulty edge e is clockwise rotated with 90° in Case 3. Next, we show that the pyramid network with one faulty edge is a pancyclic network.

Theorem 5.1. $PM_e[n]$ is pancyclic.

***Proof*:** Those links in pyramid networks are divided into three parts: layer links, external mesh-links, and internal mesh-links. In case of the faulty edge e is one of the used edges for constructing cycles in $PM[n]$, then we prove this theorem as follows:

Case 1. (e is a layer-link): By the algorithm in [21], we know that just two layer-links of two consecutive layers need to be used to construct cycles in $PM[n]$. Clearly, we can clockwise rotate the coordinate with 90, 180°, or 270 to avoid the faulty edge.

Case 2. (e is an external mesh-link): The faulty edge can be mapped to an unused external mesh-link by the mapping function τ.

Case 3. (e is an internal mesh-link): Let node $f = (k-1; u, v)$ be the common parent of the two endnodes of e. If k is 1 then the faulty edge e can be easily mapped to one of the three unused mesh-links. Otherwise, reconstruct the cycles to avoid e by removing e, adding edges $(k; x_1, y_1)(k-1; u, v)$ and $(k-1; u, v)(k; x_2, y_2)$, and then consider a faulty node f at layer k–1. □

6. Concluding Remarks

In this Paper, we developed a pair of algorithms to construct all cycles of length l, $3 \leq l \leq 5 \times 4^{k-1}$ in any two consecutive layer k and layer k–1 of the n-layer pyramid network where $2 \leq k \leq n$. By combining these two algorithms, we prove that the pyramid networks with one node or one edge fault still hold the pancyclic property unless the one-layer pyramid network with the apex fault. In other words, we can embed all cycles into pyramid networks regardless of whether there is one faulty node (edge) or not.

Acknowledgements

This work was supported in part by the National Science Council under grant NSC-91-2213-E-260-003-, Taiwan, Republic of China.

References

[1] C.C. Aggarwal, N. Jain and P. Gupta, "An efficient selection algorithm on the pyramid," *Information Processing Letters*, vol. 53, pp. 37-37, 1995.

[2] D.R. Avresky, "Embedding and reconfiguration of spanning trees in faulty hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 3, pp. 211-222, 1999.

[3] J.A. Bondy, "Pancyclic graphs," *Journal of Combinatorial Theory*, vol. 11B, pp. 80-84, 1971.

[4] R. Balakrishnan and K. Ranganathan, *A Textbook of Graph Theory*, Springer, 1999.

[5] L. Cinque and G. Bongiovanni, " Parallel prefix computation on a pyramid computer," *Pattern Recognition Letter*, vol. 16, pp. 19-22, 1995.

[6] Y.W. Chen and K.L. Chung, "A parallel approach for embedding large pyramids into smaller hypercubes with load balancing," *Journal of Information Science and Engineering*, vol. 16, pp. 117-126, 2000.

[7] F. Cao, D.Z. Du, D.F. Hsu, and S.H. Teng. "Fault tolerance properties of pyramid networks," *IEEE Transactions on Computers*, vol. 48, no. 1, 1999.

[8] R. Cipher and H.L.G. Sanz, "SIMD architectures and algorithms for image processing and computer vision," *IEEE Transactions on Acoustic, Speech, and Signal processing*, vol. 37, pp. 2158-2174, 1989.

[9] Dyer , C.R., and Rosenfeld, A., "Triangle cellular automata," *Information and Control*, vol.48, pp.54-69, 1981.

[10] Z. Guo and R.G. Melhem, "Embedding binary X-trees and pyramids in processor array with spanning buses," *IEEE Transaction on Parallel and Distributed System*, vol. 5, no. 6, pp. 664-672, 1994.

[11] S.Y. Hsieh, G.H. Chen, and C.W. Ho, "Longest fault-free paths in star graphs with vertex faults," *Theoretical Computer Science*, vol. 262, no. 1-2, pp. 215-227, 2001.

[12] S.Y. Hsieh, C.W. Ho, and G.H. Chen, "Longest fault-free paths in star graphs with edge faults," *IEEE Transactions on Computers*, vol. 50, no. 9, pp. 960-971, 2001.

[13] J.F. Jenq and S. Sahni, "Image shrinking and expanding on a pyramid," *IEEE Transactions on Parallel and Distributed System*, vol. 4, no. 4, 1993.

[14] R.S. Lo and G.H. Chen, "Embedding longest fault-free paths in arrangement graphs with faulty vertices," *Networks*, vol. 37, no. 2, pp. 84-93, 2001.

[15] M.J. Quinn, *Design Efficient Algorithms for parallel Computers*, New York, McGraw-Hill, 1987.

[16] C.K.Y. Ng, L.K.L. Pun, D.M.C. Ip, M. Hamdi, and I. Ahmad, "Embedding pyramids into 3D meshes," *Journal of Parallel and Distributed Computing*, vol. 36, pp. 173-184, 1996.

[17] E. Schmeichel and S.L. Hakimi, "A cycle structure theorem for hamiltonian graphs," *Journal of Combinatorial Theory*, vol. 45, pp. 99-107, 1988.

[18] H. Sarbazi-Azad, M. Ould-Khaoua and L.M. Mackenzie, "Algorithmic construction of hamiltonians in pyramids," *Information Processing Letters*, vol. 80, pp. 75-79, 2001.

[19] H. Senoussi and A. Saoudi, "A quadtree algorithm for template matching on a pyramid computer," *Theoretical Computer Science*, vol. 136, pp. 387-417, 1994.

[20] D.B. West, *Introduction to Graph Theory*, Second Edition, Prentice-Hall, 2001.

[21] R.Y. Wu, *Hamiltonianicity of pyramid networks*, Master Thesis, National Chi Nan University, 2001.

[22] S.M. ZHANG, "Pancyclism and bipancyclism of hamiltonian graphs," *Journal of Combinatorial Theory*, vol. 60, pp. 159-168, 1994.