Routing and Broadcasting Algorithms for the Root-Folded Petersen etworks

Ming-Bo Lin

Associate Professor, Deparment of Electronic Engineerins. National Taiwan Unive rsiry of Science and Technology. 43 Keelung Road Section 4. Taipei. Taiwan.

Grene-Eu Jan

Associate Professor, Department of Computer Science. National Taiwan Ocean University, Keelung, Taiwan.,
b0199@hpwsl.ntou.edu.tw

ROUTING AND BROADCASTING ALGORITHMS FOR THE ROOT-FOLDED PETERSEN NETWORKS

Ming-Bo Lin* and Gene-Eu Jan**

ABSTRACT

Petersen networks have been proposed as the alternatives for interconnecting highly parallel computers. The main advantages of Petersen networks are: regular topology, high scalability, and smaller diameters. In this paper, we propose a new node naming scheme for Petersen networks. Based on this scheme, two Petersen graph related networks, called root-folded Petersen network in type-I and type-II, are constructed. Furthermore, both $O(\log_{10} N)$ routing and broadcasting algorithms for the proposed networks are developed, where N is the number of nodes.

INTRODUCTION

Advanced computers are centered around the concept of parallel processing. One major way to achieve this is by integrating multiple computers through an interconnection network. The entire system performance is then determined not only by the computers but also by the underlying interconnection network. Hence, we must have a high performance interconnection network.

Various high-performance interconnection networks have been extensively studied in the literature. Among these networks, the hypercube family has been popular due to that it has several elegant properties: symmetry, regularity, high fault-tolerance, logarithmetic degree and diameter, self-routing, and simple broadcasting schemes [1,2]. Nevertheless, new networks are being proposed and analyzed with regards to their applicability and enhanced topological or performance properties. The most popular ones are those that based-on the Petersen graphs and their derivatives. These include the folded Petersen network [3,4], the folded Petersen cube network [5], and the hyper Petersen network [6].

The major features of Petersen networks are: regular topology, high scalability, smaller diameters, and less network cost than hypercubes. However, one problem of Petersen networks is the node addressing scheme for its nodes. Different routing algorithms will arise with different node naming schemes. To derive effective routing algorithms, in the paper a new addressing scheme based on mixed radix number system is proposed.

In this paper, we also propose an alternative of folded Petersen network called the root-folded Petersen network and investigate its routing and broadcasting algorithms.

The rest of the paper is organized as follows. Section 2 describes naming schemes of Petersen graph. Section 3 presents hierarchical Petersen-related networks. Section 4 describes both routing and broadcasting algorithms for the root-folded Petersen networks. Section 5 concludes this paper.

NAMING SCHEMES OF PETERSEN GRAPH

The Petersen graph $P = (V, E)$ has ten nodes with outer 5-cycle, inner 5-cycle, and five links join them. Several schemes for naming each node of this graph are possible. In this section, we describe some naming schemes and basic routing and broadcasting algorithms for it.

Naming schemes

The simplest one is that numbers the outer cycle nodes consecutively counterclockwise starting from 0

and numbers the inner cycle nodes clockwise [4], as shown in Fig. 1(a). Another way of constructing the Petersen graph is as follows. Let all the two-element subsets of {1,2,3,4,5} represent the nodes and disjoint subset pairs form the edge-set [4]. The resulting graph is depicted in Fig. 1(b). Yhe major shortcoming of above two naming schemes is that we cannot determine whether two nodes is adjacent or not just from their node addresses except storing their linkage relationship in a table in advance. Therefore, other alternatives have to be explored.

Due to that the inherent structure of the Petersen graph consists of two cycle with each having five nodes, thus, it is natural to name its nodes based on a mixed-radix number with radices 2 and 5. The radix-2 digit is used to distinct the inner-cycle from outer-cycle while the radix-5 digit names the nodes clockwise from the top one of each cycle. The resulting Petersen graph with its node naming is shown in Fig. 1(c). Let each node be denoted as (n_1, n_0), where n_1 and n_0 are the radix-2 and radix-5 digits, respectively. Based on this naming scheme, the Petersen graph can be formulated as follows.

Definition 1 (Petersen graph)

A Petersen graph has ten nodes and 15 edges and is defined as $P = (V_P, E_P)$, where $V_P = \{(d_2, d_5)|d_2 \in \{0, 1\}$ and $d_5 \in \{0, 1, 2, 3, 4\}\}$ and $E_P = \{\{(0, d_i), (0, d_j)\}|d_j = d_i + 1 \bmod 5, \forall d_i \in \{0, 1, 2, 3, 4\}\} \cup \{\{(0, d_i), (1, d_i)\}|d_i \in \{0, 1, 2, 3, 4\}\} \cup \{\{(1, d_i), (1, d_j)\}|d_i - d_j = 2 \bmod 5, \forall d_i \in \{0, 1, 2, 3, 4\}\}$.

The cost of this addressing scheme is the same as that of the conventional approach since both approaches use 4-bit for representing any node address. More precisely, each node requires 4 bits to represent its address, one bit for the radix-2 digit and three bits for the radix-5 digit.

Other naming scheme is also possible. The one shown in Fig. 1(d) is based on *m*-out-of-*n* codes. The outer cycle uses 2-out-of-5 code to denote each node while the inner cycle 1-out-of-5 code. It is easy to conclude that, from the Fig. 1(d), this naming scheme has the following properties: (1). The Hamming distance between any two adjacent nodes on the outer cycle is 4 while nonadjacent node pair is 2. (2). The Hamming distance between any two nodes on the inner cycle is 2. (3). The Hamming distance between any two adjacent outer-node and inner-node is 1 while nonadjacent node pair is 3.

The cost of this node-addressing scheme is the same as that of the Fig. 1(b) but one bit higher than that of the Fig. 1(c). In addition, this scheme has a similar properties from the view point of designing its routing and broadcasting algorithms. Therefore, in
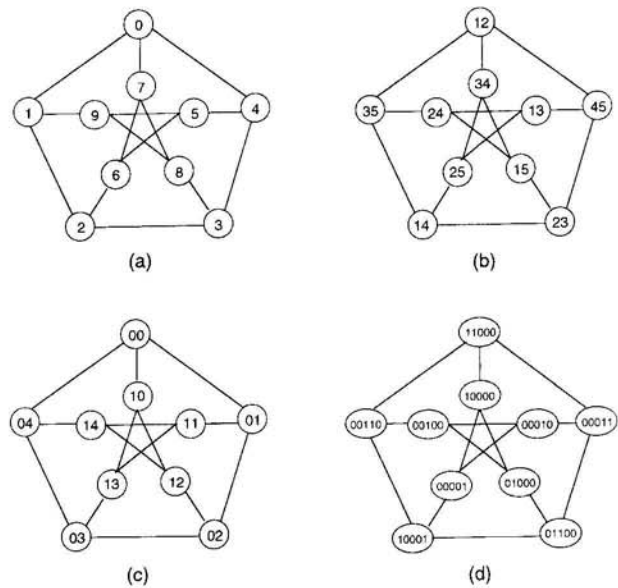

Fig. 1. The Petersen graph and its node naming schemes.

the rest of this paper, we only consider the naming method depicted in Fig. 1(c).

Basic routing and broadcasting algorithms

Based on the naming scheme for the Petersen graph described in the proceeding section, we will develop the routing and broadcasting algorithms for the Petersen graph.

For convenience, the following three neighboring node finding functions are defined.

$N_{greater}(S)$:   Return the neighboring node of S on the same cycle with node address greater than S;

$N_{smaller}(S)$:   Return the neighboring node of S on the same cycle with node address smaller than S;

$N_{changecycle}(S)$: Return the neighboring node of S on the different cycle;

In addition, one function called **adjacent** (N_1, N_2) is defined to determine whether nodes N_1 and N_2 are adjacent. Once we have these, the following routing algorithm for the Petersen network is obtained immediately.

Algorithm Basic-Routing(S, D)

{Assume that the source and destination node addresses are represented as $S(n_1, n_0)$ and $D(n_1, n_0)$, respectively, where n_1 is a radix-2 digit and n_0 is a radix-5 digit.}

Begin

    while $D \neq S$ **do**

if adjacent (D, $S_{new} = N_{greater}(S)$) then forward to S_{new}.

else if adjacent (D, $S_{new} = N_{smaller}(S)$) then forward to S_{new}.

else if adjacent (D, $S_{new} = N_{changecycle}(S)$) then forward to S_{new}.

Forward from S_{new} to D.

end {of algorithm Basic-Routing}

As it can be seen from the naming scheme depicted in Fig. 1(c), all of neighboring finding functions $N_{greater}(S)$, $N_{samller}(S)$, and $N_{changecycle}(S)$ are easily to be designed. Thus, they are omitted here. As for the function **adjacent**(N_1, N_2), several cases have to be considered: both nodes are on the same cycle or on the different cycles. It is described in a more formal way as follows.

Algorithm adjacent(N_1, N_2)

{Both nodes N_1 and N_2 are represented as a mixed radix pair:(n_1, n_0) where n_1 is a radix-2 digit and n_0 is a radix-5 digit.}

Begin

case of ($N_1(n_1)$, $N_2(n_1)$) do

$N_1(n_1) = N_2(n_1)$: {Both nodes are on the same cycle.}

if ($N_1(n_1) == 0$ and ($N_1(n_0) - N_2(n_0)$) mod 5 == 1) then return true

else if (($N_1(n_0) - N_2(n_0)$) mod 5 == 2 or 3) then return true

$N_1(n_1) > N_2(n1)$: {N_1 is on the inner cycle while N_2 on the outer cycle.}

if ($N_1(n_0) == N_2(n_0)$) then return true

$N_1(n_1) < N_2(n_1)$: {N_2 is on the inner cycle while N_1 on the outer cycle.}

if ($N_1(n_0) == N_2(n_0)$) then return true

end {of case}

return false {Nodes N_1 and N_2 are nonadjacent.}

end {of algorithm adjacent}

In addition to basic routing algorithm, the capability of broadcasting a message from any node to all other nodes is also an important feature of any interconnection network. Therefore, in what follows we consider how to broadcast a message from any node to all other nodes in the Petersen graph.

Algorithm Basic-Broadcasting(S)

{The source node address is represented as $S(n_1$, $n_0)$, where n_1 is a radix-2 digit and n_0 is a radix-5 digit.}

Begin

Forward to all its neighboring nodes: $N_{greater}(S)$, $N_{samller}(S)$, and $N_{changecycle}(S)$.
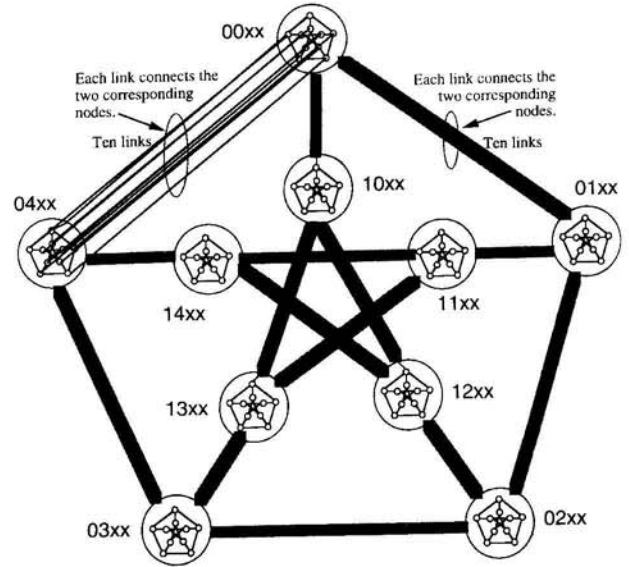
Forward to all the neighboring nodes of

Fig. 2.   A schematic representation of the two-dimensional folded Petersen graph (FP_2).

$N_{greater}(S)$, $N_{samller}(S)$, and $N_{changecycle}(S)$ which did not receive the message.

end {of algorithm Basic-Broadcasting}

It is easy to show that both **Basic-Routing** and **Basic-Broadcasting** algorithms have constant time complexities and dead-lock free since they use disjoint nodes.

HIERARCHICAL PETERSEN-RELATED NETWORKS

To generalize the Petersen graph to n dimensions, several schemes have been proposed. Among these the following one called folded Petersen graph [3, 4] is the most pronounced due to that it has the node- and edge-symmetric properties.

The folded Petersen graph

The formal definition of the folded Petersen graph in terms of the above node naming scheme is given as follows.

Definition 2 (Folded Petersen Graph)

An n-dimensional folded Petersen graph is defined as $FP_n = (V_{FP_n}, E_{FP_n})$, where $V_{FP_n} = \{(D_n, D_{n-1}, ..., D_1)|D_i \in V_P\}$ and $E_{FP_n} = \{\{(D_n, D_{n-1}, ..., D_i, ..., D_1), (S_n, S_{n-1}, ..., S_i, ..., S_1)\}|D_j = S_j, \forall j \neq i$ and $(D_i, S_i) \in E_P$, for $1 \leq i, j \leq n\}$.

An example of the two-dimensional folded Petersen graph is shown in Fig. 2. The interconnecting network of parallel computer systems based on

Table 1. Tological properties comparison of various hierarchical Petersen Graphs

Tology	# Nodes N	Diameter d	Links l	Cost $c = d \cdot \delta$	Degree δ	Connectivity k
Root-Folded Petersen (RFP_n-I)	10^n	$4n-2$	$15(10n-9)$	$12n^2-6n$	root=$3n$ other=3	3
Rooted-Folded Petersen (RFP_n-II)	10^n	$4n-2$	$15(10n-9)$	$12n^2-6n$	superroot =$n+2$ other=3	3
Folded Petersen (EP_n)	10^n	$2n$	$\frac{19}{6}10n-\frac{50}{3}$	$6n^2$	$3n$	$3n$
Folded Petersen Cube ($EPQ_{m,n}$)	$2^m 10^n$	$m+2n$	$25^n(m+1)2^m$	$m^2+5mn+6n^2$	$m+3n$	$m+3n$
Hyper Petersen (HP_{m+3n})	$\frac{5}{4}2^m 8^n$	$m+3n-1$		$(m+3n)\times(m+3n-1)$	$m+3n$	$m+3n$

the Petersen graph is called the Petersen network while on the folded Petersen graph called the folded Petersen network.

Some important topological properties of the folded Petersen graph and the other Petersen-related graphs are listed in Table 1. As can be seen from the table, the links of an *n*-dimensional FP_n network is proportional to 10^n, which may be too large to be practical in some systems. To alleviate this limitation, at least two alternatives may be used. They are based on the observation that the number of links in the FP_n networks increase exponentially due to the fact that the node degree is $3n$, that is, every node is connected to other corresponding nodes in different dimensions. As a consequence, the number of links is reduced significantly if allow this linking to the root node only (i.e., node address is 00 ... 0).

The root-folded Petersen graph

Two different architectures can then be formed. The first one is called the root-folded Petersen graph in type-I and is defined in terms of the basic Petersen graph node naming scheme.

Definition 3 (Root-Folded Petersen Graph-Type I)
An *n*-dimensional root-folded Petersen graph in type-I is defined as $RFPI_n = (V_{RFPI_n}, E_{RFPI_n})$, where $V_{RFPI_n} = \{(D_n, D_{n-1}, ..., D_1)|D_i \in D_P\}$ and $E_{RFPI_n} = \{\{(D_n, D_{n-1}, ..., D_i, ..., D_1), (S_n, S_{n-1}, ..., S_i, ..., S_1)\}|(D_1, S_1) \in E_P$, or $(D_i, S_i) \in E_P \, \forall 2 \le i \le n$ and $S_j = D_j = 00$, $\forall j < i\}$.

An example of this kind network is shown in Fig. 3. The major disadvantage of this network is that the reliability property strongly depends on the availability of root node. That is, the network is disconnected if one root node of it fails. Therefore, a
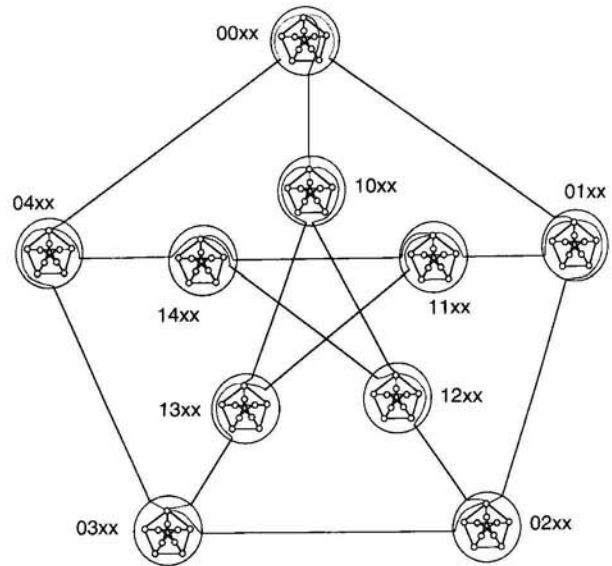

Fig. 3. A schematic representation of the two-dimensional rooted-folded Petersen graph (RFP_2-I).

variation of above network is defined in the following. Due to the similarity between them, this new one is called the root-folded Petersen graph in type-II. An n-dimensional root-folded Petersen graph in type-II is defined as $RFPII_n = (V_{RFPII_n}, E_{RFPII_n})$, where $V_{RFPII_n} = \{(D_n, D_{n-1}, ..., D_1)|D_i \in V_P\}$ and E_{RFPII_n} contains the same links as E_{RFPI_n} except that the root nodes are replaced by supernodes consisting of three nodes: 00, 01, and 04. An example of this type network is shown in Fig. 4. Due to the similarity between both types of the root-folded Petersen graphs, for the rest of this paper we will concentrate our efforts on the routing and broadcasting algorithms for the root-folded Petersen graph in type-I only.
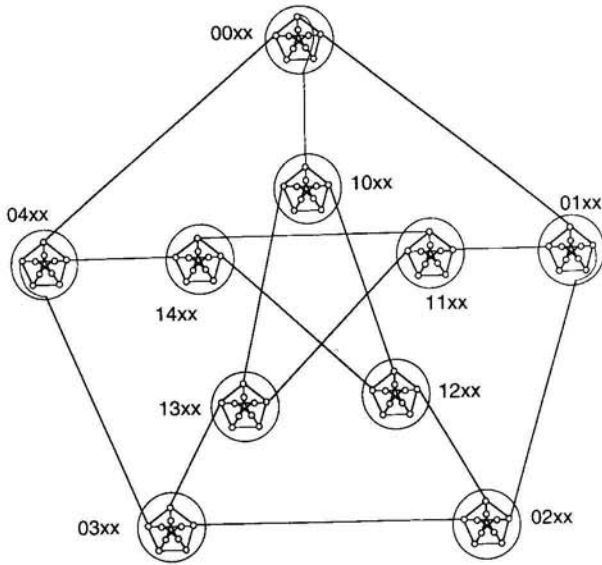
Fig. 4. A schematic representation of the two-dimensional root-folded Petersen graph (RFP_2-II).

ROUTING AND BROADCASTING ALGORITHMS

In this section, we describe both the routing and broadcasting algorithms for the root-folded Petersen networks. In addition, two spanning trees: downward and upward are defined.

Routing algorithms

Once we have the Basic-Routing algorithm, the routing problem of folded Petersen network can be easily solved as follows.

Algorithm FP_n-Routing(S, D)
{The source and destination node addresses are represented as $S(S_nS_{n-1}...S_1)$ and $D(D_nD_{n-1}...D_1)$, respectively, where S_i and D_j are a pair of mixed-radix digits and represented as (n_1, n_0).}
Begin
  for $i = n$ **to** 1 **step** −1 **do**
    Basic-Routing(S_i, D_i);
  end {of for}
end {of algorithm FP_n-Routing}

Similarly, the routing problem of root-folded Petersen network can also be easily solved based on the Basic-Routing algorithm.

Algorithm RFP_n-Routing(S, D)
{The source and destination node addresses are represented as $S(S_nS_{n-1}...S_1)$ and $D(D_nD_{n-1}...D_1)$, respectively, where S_i and D_j are a pair of mixed-radix

digits and represented as (n_1, n_0).}
Begin
  $i = n$;
  while $D_i = S_i$ **do** $i = i−1$;
    if $i = 0$ **then break**; {S and D are the same node.}
    else begin
      for $j = 1$ **to** i **step** 1 **do** Basic-Routing(S_i, 00);
      for $j = i$ **to** 1 **step** −1 **do** Basic-Routing(00, D_i);
    end {of else}
  end {of while}
end {of algorithm RFP_n-Routing}

It is easy to see that both algorithms mentioned above have time complexities of $O(n)$, where n is the dimension of the underlying network. In addition, the paths generated by these algorithms are node-disjoint. Hence, they are dead-lock free.

Broadcasting algorithms

Once we have the Basic-broadcasting algorithm, the broadcasting problem of folded Petersen network can be easily solved as follows.

Algorithm FP_n-Broadcasting(S)
{The source address is represented as $S(S_nS_{n-1}...S_1)$, where S_i is a pair of mixed-radix digits and represented as (n_1, n_0).}
Begin
  for $i = n$ **to** 1 **step** −1 **do**
    Basic-Broadcasting(S_i);
  end {of for}
end {of algorithm FP_n-Broadcasting}

Similarly, the broadcasting problem of folded Petersen network can be easily solved based on the described Basic-Broadcasting algorithm. The resulting broadcasting algorithm for RFP_n networks is described as follows.

Algorithm RFP_n-Broadcasting(S)
{The source address is represented as $S(S_nS_{n-1}...S_1)$, where S_i is a pair of mixed-radix digits and represented as (n_1, n_0).}
Begin
  for $i = 1$ **to** n **step** 1 **do** Basic-Routing(S_i, 00);
  for $i = n$ **to** 1 **step** −1 **do**
    Basic-Broadcasting(S_i)
  end {of for}
end {of Algorithm RFP_n-Broadcasting}

It is easy to see that both algorithms mentioned above have time complexities of $O(n)$, where n is the dimension of the underlying network. In addition, the

(a) Downward spanning tree   (b) Upward spanning tree
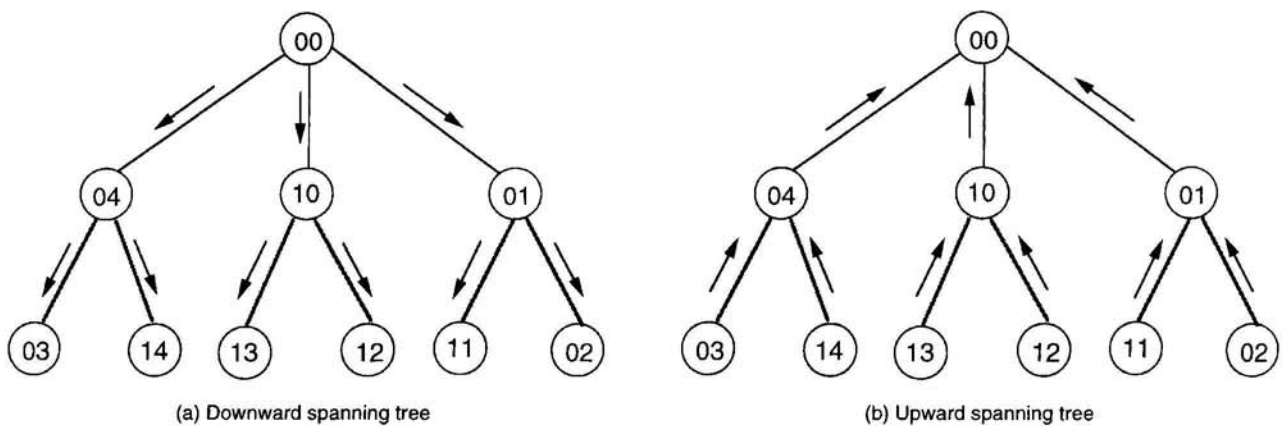
Fig. 5. The spanning trees: downward and upward, of the Petersen network.

paths generated by these algorithms are node-disjoint. Hence, they are dead-lock free.

The tracks of RFP_n-**Broadcasting**(S) algorithm forms a downward spanning tree if $S(S_nS_{n-1}...S_1) = 00...0$. An example of RFP_1 is shown in Fig. 5(a). The collection of reverse tracks of a downward spanning tree is defined as the upward spanning tree. Fig. 5(b) depicts one such tree for Fig. 5(a).

CONCLUSION

In this paper, we proposed a naming scheme as well as basic routing and broadcasting algorithms for Petersen network. Based on this naming scheme, two hierarchical Petersen networks called root-folded Petersen network type-I and type-II are constructed. Furthermore, both $O(\log_{10} N)$ routing and one-to-all broadcasting algorithms for the root-folded Peter network are developed. In addition, two minimum spanning trees are explored.

REFERENCES

1. Hwang, K., *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, New York (1993).
2. Kumar, V., Grama, A., Gupta, A. and Karypis, G., *Introduction to Parallel Computing: Design and Analysis of Algorithms*, The Benjamin/Cummings Publishing Company, Inc., Pedwood City, CA (1994).
3. Shen, H. "A High Performance Interconnection Network for Multiprocessor Systems," *Parallel Computing*, Vol. 19, pp. 993-1001 (1993).
4. Öhring, S. and Das, S.K. "The Folded Petersen Network: A New Communication-Efficient Multiprocessor Topology," *1993 International Conference on Parallel Processing*, Vol. I, pp. 311-314, Aug. (1993).
5. Öhring, S. and Das, S.K., "Folded Petersen Cube Networks: New Competitors for the Hypercubes," *IEEE Trans. on Parallel and distributed Systems*, Vol. 7, No. 2, pp. 151-168 (1996).
6. Das, S.K., Öhring, S. and Banerjee, A.K., "Embeddings into Hyper Petersen Networks: Yet Another Hypercube-Like Interconnection Topology," *Journal of VLSI*, special issue on interconnection networks, Vol. 2, No. 4, pp. 335-351 (1995).

根摺疊式彼得生網路路由與廣播演算法

林　銘　波

國立台灣科技大學電子工程系

詹　景　裕

國立台灣海洋大學資訊科學系

摘　要

彼得生網路由於具有下列重要特性：規則結構、高可塑性、與小網路直徑等，近年來頗受研究學者的重視。本論文中，除了對彼得生網路的節點命名方法做一研究之外，也提出一種新的彼得生網路，稱為根摺疊式彼得生網路，並且在此網路上發展出$O(\log N)$的路由與廣播演算法。