ON NON-BLOCKING PROPERTIES OF PARALLEL DELTA NETWORKS

F. Bernabei, A. Forcina*, M. Listanti

Fondazione Ugo Bordoni - Roma (Italy)

ABSTRACT

This paper deals with the definition of a class of NxN interconnection networks called Parallel Delta Network (PDN). For this class of networks the non-blocking conditions are given. In particular, by means of the graph colouring technique, it has been proved that the minimum number of Delta subnetworks (L) necessary to provide the non-blocking property is $L=n^{\lceil S/2 \rceil -1}$ where n is the size of the basic switching element and S the number of stages required by an NxN Delta network. A routing algorithm for the establishment of any permutation has been defined. It operates for any value of n and shows a polynomial time complexity equal to $O(N^{3/2})$. Moreover, in case of the set-up of a single connection request, this algorithm assures a time complexity equal to $O(\sqrt{N})$. This property makes it well suitable to an asynchronous telecommunication environment.

1. Introduction

Within the last years Delta networks have raised a large interest in parallel computing environment [1,2,3]; moreover, due to their self-routing property, they appear very promising in designing high throughput switching architectures [4,5,6]. In both these environments the non–blocking or rearrangeable features[1] are of paramount importance to efficiently handle the incoming data streams.

On the contrary Delta networks show an intrinsic constraint given by the existence of a single path between a generic couple of inlets/outlets. This property involves a low throughput [7,8] and no fault tolerance capability [9,10]. In order to reach the non-blocking property, two main alternatives are possible: i) addition of redundant stages to a Delta network; ii) assembling Delta networks in parallel.

The first approach has been followed by Benes [11], who defined a rearrangeable network which is obtained by merging two Baseline networks. An NxN Benes network provide N multiple input/output paths and its routing algorithm [12,13,14] presents a time complexity equal to $O(N \cdot \log_2 N)$. More recently, Lee [15] proved that the property of rearrangeability takes validity for any $2 \cdot (\log_2 N)-1$ stage networks given by the merging of two Delta networks.

The drawback of this approach is represented by the large number of stages which composes a network and consequently by the high transit delay. This latter aspect is emphasized in the Batcher network [16]. This topology in series with a Delta network achieves rearrangeability, but it implies a very large number of stages (i.e. $1/2 \cdot (\log_2 N) \cdot (\log_2 N+1)+\log_2 N$). This architecture is therefore unsuitable for applications requiring a high value of N.

As for the assembling of Delta networks in parallel, performance aspects for fault-tolerance as well as throughput and delay, have been discussed in [8,17,18], whereas the problem of performing arbitrary permutations has been studied in [19,20]. In first paper the upper bounds for the number of Omega networks (L), composed of 2x2 Switching Elements (SEs), required to perform a generic permutation is given ($L=2^{\lfloor 1/2 \cdot \log_2 N \rfloor}$). In [20] a class of networks, called INDRA, is presented and an algorithm for realizing the Bit Permute Class (BPC) of permutation is described.The authors claim that the problem of defining a general procedure for realization of an arbitrary permutation is still unsolved.

In this paper a new class of networks, called Parallel Delta Network (PDN), is presented and in spite of [20] the conditions which guarantee the rearrangeability property are given. These conditions are not constrained by any value of the network size as well as SE size. Moreover, an

[1] As non-blocking property we define the capability of a network to establish an arbitrary permutation of the inlets onto the outlets. Rearrangeability refers to the same capability, but the establishment of a single connection can require rerouting of already set-up paths.

Work carried out in the framework of the agreement between Fondazione Ugo Bordoni and the Italian PT Administration.

*A. Forcina is now with S.I.P. Headquarters, Roma, Italy.

CH2534-6/88/0000-0326 $1.00 © 1988 IEEE

algorithm which is able to realize any arbitrary permutation is discussed.

In the following the study is based on a PDN composed of Baseline networks. However the obtained results, in particular the rearrangeability conditions and the routing algorithm, hold for any type of PDN. That is guaranteed by the well-known topological equivalence among Delta networks [2,3].

The analysis is based on the very simple concept of distance between two links coming to the same stage and, by using graph colouring techniques, it leads to the condition on the number of Delta subnetworks required in order to achieve the non-blocking property.

The routing algorithm has been designed with the aim of satisfying two different requirements: i) the minimization of the time complexity for the establishment of a complete permutation; ii) the minimization of the time complexity for the set-up of a single connection request, whatever network state is, with the minimum number of rerouted paths.

The first requirement has been satisfied by proposing an algorithm (the Permutation Routing Algorithm) with a polynomial time complexity equal to $O(N^{3/2})$. The latter requirement, which is typical of an asynchronous telecommunication environment, has been met by assuring the set-up of a single connection request with time complexity equal $O(\sqrt{N})$. Moreover the maximum number of rerouted paths is lesser than $2\cdot\sqrt{N}$.

Next section is devoted to some preliminaries definitions. In Sec. 3 the PDN and its properties are presented, whereas the non-blocking conditions are discussed in Sec. 4. The routing algorithm is described in Sec. 5 where its relevant complexity is also computed. Sec. 6 summarizes the achieved results.

2. Preliminaries

A Baseline Delta network can be identified with the triple (N,n,S) where N is the network size (i. e. the number of inlets and outlets labelled from 0 to N-1), n is the size of the basic SE and S is the number of the switching stages ($S=\log_n N$) labelled from 1 to S.

In this framework, it is possible to define the distance between two links.

Def 1: the distance $d_{in}^{n,s}(x,y)$ ($d_{out}^{n,s}(x,y)$) between two links (x,y) ingoing to (outgoing from) the stage s of a network with SE nxn is given by

$$\max z \mid \lfloor x/n^z \rfloor \neq \lfloor y/n^z \rfloor \quad (0 \leq x,y \leq N-1) \qquad (1)$$
$$\text{with } z = 0,1,2,....$$

It is worth noting that we shortly indicate with $d_{in}(x,y)$ ($d_{out}(x,y)$) the distance between two given links x and y ingoing to (outgoing from) the network and with d_{in} (d_{out}) if the distance is calculated between two not specificated inlets (outlets).

This definition implies that the n links connected to the same SE are set at a distance of 0 to each other, whereas the link 0 and the link N-1 are set at a distance equal to S–1.

The following properties concerning the physical meaning of distance can be straightforwardly derived:

Property 1: let two selected inlets of a (N, n, S) Baseline network be located at distance d_{in}. Any pair of paths sourcing from these inlets either do not share any links or the first shared link is the one outgoing from the merging SE at stage $s=d_{in}+1$.

Property 2: let two selected outlets of a (N, n, S) Baseline network be located at distance d_{out}. Any pair of paths directed towards these outlets either do not share any links or the last shared link is the one ingoing to the splitting SE at stage $s=S-d_{out}$.

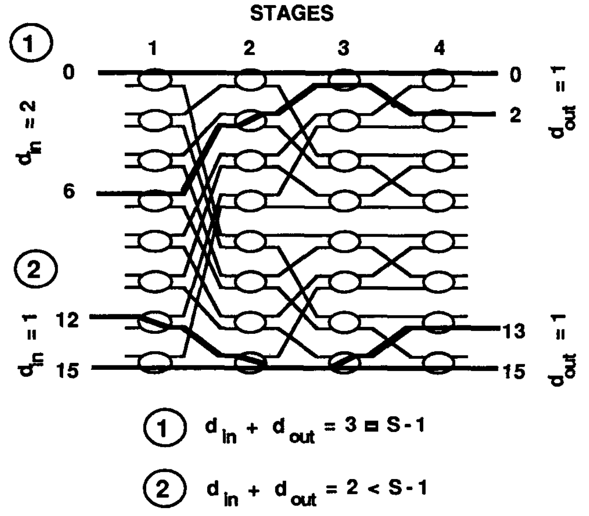


Fig.1 - A representation of the non-blocking condition for a Baseline network.

The previous properties lead to the following theorem which provides the non-blocking condition for a pair of paths in a Baseline network:

Theorem 1: let two selected inlets of a (N,n,S) Baseline network be at distance d_{in} and to be connected to two generic outlets at distance d_{out}, respectively. The two paths are disjoint (i.e. they do not share any link) if and only if

$$d_{in}+d_{out} \geq S-1. \qquad (2)$$

Proof: on the basis of the properties 1 and 2 if the requested paths must be disjoint the splitting stage must be anterior or coincident with respect to the merging stage. In the anterior case the paths are obviously disjoint, whereas in the coincident case no conflict can arise because the paths are routed towards different outgoing links. Therefore, $S-d_{out} \leq d_{in}+1$ or $d_{in}+d_{out} \geq S-1$. □

Fig.1 shows a graphic representation of the non-bloking condition presented in theorem 1.

3. The Parallel Delta Network

An NxN Parallel Delta Network (PDN) is built up by : i) L Delta networks (N,n,S) in parallel, called layers; ii) one additional switching stage (labelled with 0) composed of nxn SEs; iii) two complementary stages, called EXP and CNC, which perform the expansion and concentration functions, respectively.

The first two items define an LNxLN interconnection network which is called Kernel and labelled with the triple (LN,n,S+1), wherein the three parameters have the previously stated meaning. In particular, if L=n the Kernel is an LNxLN Delta network.

In order to univocally define a PDN, the connection rules associated to the following interstage bundles must be given:

B1) stage EXP and the stage 0 of the Kernel;
B2) the stages 0 and 1 of the Kernel;
B3) the stage S of the Kernel and the CNC stage.

The definition criterium of these bundles has the aim of maximizing, at the inputs of stages 0 and 1, the distance between the multiple paths sourcing from any PDN inlet/outlet pair. In the following discussion, these rules are designed for the Baseline network; similar rules can be easily found for different Delta networks.

Let i ($0 \leq i \leq LN-1$) be an outgoing link from a generic stage and j ($0 \leq j \leq LN-1$) an ingoing link to the successive stage. The link i has to be connected to link j according to the following rules:

B1) $$j = \lfloor i / L \rfloor + (i \text{ MOD } L) \cdot N \qquad (4)$$

B2) $$j = \lfloor i / n \rfloor + (i \text{ MOD } n) \cdot LN/n. \qquad (5)$$

B3) $$j = \lfloor i / N \rfloor + (i \text{ MOD } N) \cdot L \qquad (6)$$

Fig.2 shows a general represenation of a PDN.

It is evident that a PDN provides P=L·n different paths within the Kernel between a generic inlet/outlet pair. Each path is identified by a "PDN address" which is composed of two fields: i) "distribution address"; ii) "destination address".

The "distribution address" is composed of two digits L–radix and n-radix, respectively; the first one provides routing through the stage EXP, the second one through the stage 0. The "destination address" is an S digit n-radix pattern which provides routing through a layer of the Kernel. Fig.3 exhibits a sketched representation of the "PDN address".

A generic PDN is characterized by the following properties under the assumption of the previous definition of distance between links

Property 3: let a PDN have L layers. Among the P paths connecting an inlet/outlet pair, n out of them reach a same layer at ingoing links placed at the maximum distance from each other. Such a distance equals S-1.

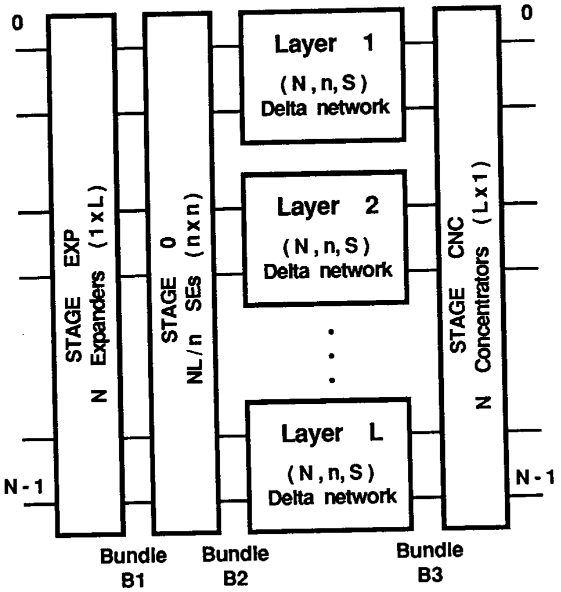


Fig. 2 - A general representation of a PDN.

PDN Address		
Distribution Address		Destination Address
1 digit L radix (Stage EXP)	1 digit n radix (Stage 0)	S digits n radix (Stages 1,2,...,S)

Fig.3 - The PDN Address bit pattern.

The property is straightforwardly derived by observing the permutation performed by the bundles B1 and B2. Fig.4 shows the P paths connecting an inlet/outlet pair.

Lemma 1: in a PDN, given P connection requests between P inlet/outlet pairs, their relevant paths do not share any link if they are routed with P different distribution addresses.

Proof: the distribution addresses of any pair of connection requests are different, hence no conflict can occur in the stage 0. As for routing within the Kernel only two cases are possible: in the first case, their relevant paths pass through different layers so that they are obviously disjoint; in the second case, these paths are addressed towards the same layer. This case implies that both the second digits of the distribution addresses are equal and the first ones are different. Under these conditions, due to the permutation performed by the bundles B1 e B2, the n paths routed towards the same layer reach incoming links of the stage 1which are set at distance S-1. Hence, on the basis of theorem 1 the P paths do not share any link. ◻

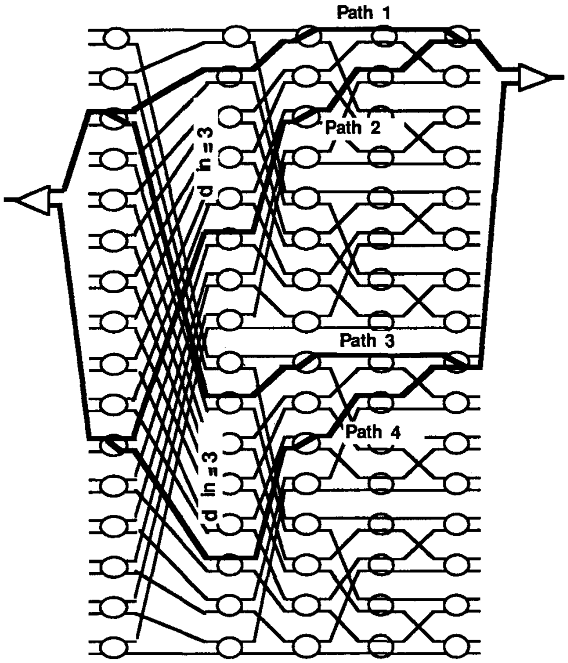


Fig.4 - The P = L·n paths between an inlet/outlet pair in a PDN.

Moreover, it is possible to give a theorem which provides the general condition for two paths not sharing any link in a PDN.

Theorem 2: let, in a PDN, two connection requests be such that their inlets are located at a distance d_{in} and their outlets at a distance d_{out}. Their relevant paths are disjoint (i.e. they do not share any links) if and only if either the two distribution addresses are different or the distribution addresses are equal and the following relation is satisfied

$$d_{in}+d_{out} \geq S. \qquad (7)$$

Proof: if the two distribution addresses are different this theorem is equivalent to property 4. If the two distribution addresses are equal, the permutations performed by the bundles B1 and B2 assure that the two paths reach links, ingoing to the same layer, which are set at distance d_{in}-1. (Note that if d_{in}=0 the two paths share a link outgoing from the stage 0).

Dually, due to the permutation accomplished by the bundle B3, two outlets at a distance d_{out} correspond to two outgoing links from the stage S which are set at the same distance d_{out}.

By applying theorem 1, we have $d_{in}-1+d_{out} \geq S-1$ or equivalently the relation (7) ◻

4. The non-blocking conditions

In the previous sections we have claimed the property of rearrangeability for a PDN with a required number of layers. The following theorem proves this property.

***Theorem 3*:** a PDN is rearrangeable if and only if its number of layers (L) is greater or equal to $n^{\lceil S/2 \rceil -1}$ or, equivalently, the number of paths (P) within the Kernel between a generic inlets/outlets pair is greater or equal to $n^{\lceil S/2 \rceil}$; in formulas:

$$L \geq n^{\lceil S/2 \rceil -1} \qquad (8)$$

$$P \geq n^{\lceil S/2 \rceil}. \qquad (9)$$

***Proof*:** the condition is necessary: let a number of layers be L' less than L (e.g. L'=L-1) be required to set-up any permutation. It can be proved that at least a permutation exists such that L' layers, or equivalently P'=L'·n, are not sufficient to establish it.

Let i_l and o_l ($0 \leq l \leq N-1$) denote the inlet and the outlet of the l-th connection request, the Identity permutation is defined as a permutation such that $i_l=o_l=l \ \forall \ l$.

The first P' ($0 \leq l \leq P'-1$) connection requests must be routed with P' different distribution addresses otherwise, on the basis of relation (7) of theorem 2, their relevant paths would give rise to conflicts. Hence, the (P'+1)-th connection request has to be routed with a previously employed distribution address. The maximum values of d_{in} and d_{out} ($d_{in}=d_{out}$ in the Identity permutation) computed between inlets and outlets, respectively, of the connection request to be establish and the P' already set-up paths, is equal to:

$$d_{inmax} = d_{outmax} = \lceil S/2 \rceil -1 \qquad (10)$$

and so,

$$d_{inmax}+d_{outmax} = 2 \cdot \lceil S/2 \rceil - 2 < S \qquad (11)$$

Therefore, L'=L-1 layers are not sufficient to set-up the Identity permutation.

The condition is sufficient: to prove that $L \geq n^{\lceil S/2 \rceil -1}$ layers guarantee the rearrangiability of a PDN, on the basis of property 4 and theorem 2, it is sufficient to verify that, for any permutation, relation (7) holds for any pair of paths characterized with the same distribution address. This goal can be also achieved by referring, instead of (7), to the following more restrictive conditions:

$$d_{in} \geq \lceil S/2 \rceil \qquad (12a)$$

$$d_{out} \geq \lceil S/2 \rceil \qquad (12b)$$

In order to prove that, it is convenient to define a non–directed graph G, called "Permutation Conflict Graph", in which each vertex represents a connection request and an edge exists between two vertices if and only if expressions (12) do not hold. Formally:

$$G=[V, E]$$

in which V is the set of vertices

$$V=\{v_h; \ h=0,1,....,N-1\}$$

wherein v_h is the h-th connection request between the inlet i_h and the outlet o_h

$v_h=(i_h,o_h)$ $h=0,1,..,N-1.$

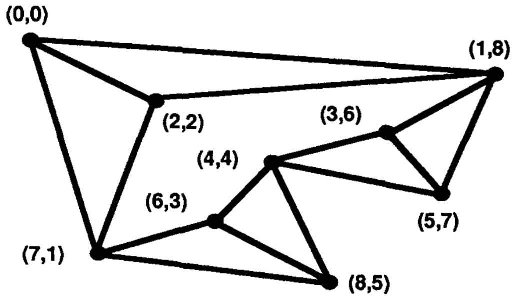
and E is the set of edges

$E=E_{in} \cup E_{out}$

$E_{in}=\{(v_h,v_k) \mid d_{in}(i_h,i_k) < \lceil S/2 \rceil,\ h<k\}$

$E_{out}=\{(v_h,v_k) \mid d_{out}(o_h,o_k) < \lceil S/2 \rceil,\ h<k\}$

Fig.5a depicts an example of a permutation and its relevant Permutation Conflict Graph.



Permutation to be routed

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 8 & 2 & 6 & 4 & 7 & 3 & 1 & 5 \end{pmatrix} \begin{matrix} \text{Inlets} \\ \text{Outlets} \end{matrix}$$

Fig.5a - Example of a permutation and its relevant Permutation Conflict Graph for a PDN with N=9 and n=3.

The previous definition of G implies that an edge exists between two vertex if and only if $d_{in}(i_h,i_k) < \lceil S/2 \rceil$ or $d_{out}(o_h,o_k) < \lceil S/2 \rceil$. In other words, since all the couples of connection requests which do not satisfy expression (12) must be routed with different distribution addresses, their relevant vertices in the graph are linked by an edge. Consequently, the problem of assigning one distribution address out of P to each connection request is equivalent to colour the graph G with P colours so that two adjacent vertices do not bear the same colour.

The graph G can be considered as the union of two graphs $G_{in}=[V, E_{in}]$ and $G_{out}=[V, E_{out}]$ both composed of N/P disjoint cliques[2], whose cardinality equals P. Consequently, the clique number $\omega(G)$[3] is also equal to P. These features can be justified by observing that all and only the couples of vertices $[v_h=(i_h,o_h)\ ,\ v_k=(i_k,o_k)]$ such that

$$j \cdot P \le i_h,i_k < (j+1) \cdot P \qquad j=0,1,..,N/P-1 \qquad (13a)$$

or

$$j \cdot P \le o_h,o_k < (j+1) \cdot P \qquad j=0,1,..,N/P-1 \qquad (13b)$$

are linked by an edge each other.

Fig. 5b shows an alternative representation of the Permutation Conflict Graph (presented in fig. 5a) in which the edges are not drawn and the cliques have been emphasized.

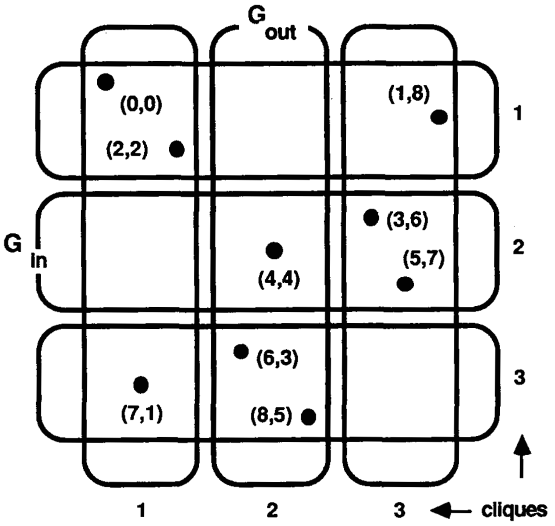


Fig.5b - An alternative representation of the Permutation Conflict Graph presented in fig.5a in which cliques are emphasized.

The graph G is a "perfect graph" [21,22]. In fact, on the basis of the Strong Perfect Graph Conjecture[4] (SPGC) [21,22], whatever chosen an odd cycle on the graph, this cycle contains at least two adjacent edges belonging to the same clique (fig.6) and, consequently, a chord always exists in the cycle.

As the graph G is perfect, the chromatic number $\gamma(G)$[5] equals $\omega(G)$ [21,22], and conseguently $\gamma(G)=P$. So $P=n^{\lceil S/2 \rceil}$ colours are sufficient to colour the graph G. As a result, all the couples of connection requets which do not satisfy relations (12) can be routed with different

[2] Let $\mathcal{G}=[\mathcal{V},\mathcal{E}]$ be, a clique is a subset of $\mathcal{V}$ that induces a complete subgraph on $\mathcal{G}$ [21,22].

[3] Let $\mathcal{G}=[\mathcal{V},\mathcal{E}]$ be, the clique number $\omega(\mathcal{G})$ is equal to the cardinality of the maximal clique di $\mathcal{G}$ [21,22].

[4] The Strong Perfect Graph Conjecture claims the "perfection" of a graph $\mathcal{G}=[\mathcal{V},\mathcal{E}]$ if and only if $\mathcal{G}$ does not contain a subset $\mathcal{A}$ of $\mathcal{V}$ such that $\mathcal{G}_{\mathcal{A}}$ or its complementary graph is an odd cycle of lenght greater than 3 without chords [21,22].

[5] Let $\mathcal{G}=[\mathcal{V},\mathcal{E}]$ be, the chromatic number $\gamma(\mathcal{G})$ is the minimum numbers of colours required to label all the vertices of $\mathcal{G}$ such that no two adjacent vertices bear the same value[21,22].

distribution addresses, and $L=n^{\lceil S/2\rceil -1}$ layers are sufficient to make a PDN rearrangeable. □

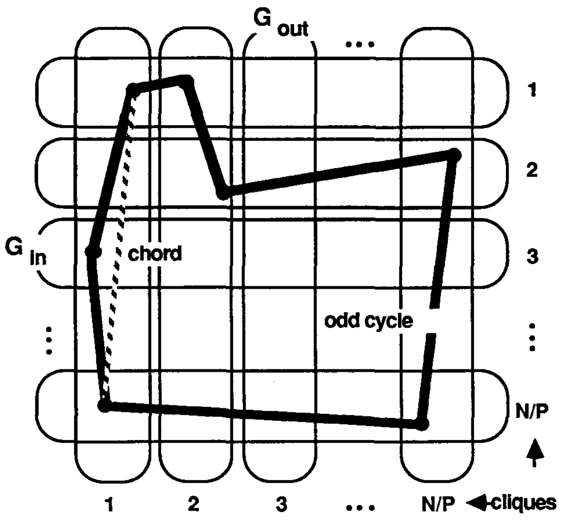


Fig.6 - *An odd cycle in a Permutation Conflict Graph.*

5. The routing algorithm

In the previous paragraphs we have defined the structure of a PDN and given its main properties. In particular the minimum number of layers to reach the network rearrageability has been found.

In this paragraph an efficient routing algorithm will be discussed. It operates for any value of the SE size and is particularly suitable in a telecommunication enviroment.

The routing algorithm is based on a step by step colouring of the N vertices of G. At each step j, the algorithm considers only the subgraph G^j identified by the vertex to be coloured and the j-1 vertices previously labelled and it admits the recolouring of some of these latter ones.

In order to describe the routing algorithm, the following definitions are necessary:

i) let $C_{in}^j(v)$ and $C_{out}^j(v)$ be two cliques in G^j located by the vertex v and belonging to G_{in} and G_{out}, respectively.

ii) let $AC[P, C_{in}^j(v)]$ ($AC[P, C_{out}^j(v)]$) be a function which provides the set of colours out of P not utilized to label the subgraph induced by $C_{in}^j(v)$ ($C_{out}^j(v)$); indeed, the subgraphs induced by $C_{in}^j(v)$ and $C_{out}^j(v)$ contain uncoloured vertices.

The routing algorithm is sintetically expressed as follows:

Permutation Routing Algorithm (nxn SE)

```
PROCEDURE Permutation Conflict Graph Colouring
FOR j=0 TO N-1 DO
    CHOOSE an uncoloured vertex v
    COMPUTE $AC[P,C_{in}^j(v)]$ and $AC[P,C_{out}^j(v)]$
    IF a colour $c \in AC[P,C_{in}^j(v)] \cap AC[P,C_{out}^j(v)]$ exists
    THEN
        ASSIGN the colour c to the vertex v
    ELSE
        SELECT one colour $c' \in AC[P,C_{in}^j(v)]$ and
                  one colour $c'' \in AC[P,C_{out}^j(v)]$
        ASSIGN the colour c' to the vertex v
        v' = v
        WHILE a vertex $w \in C_{out}^j(v')$ with colour c' exists DO
            v" = w
            ASSIGN the colour c" to the vertex v"
            IF a vertex $w \in C_{in}^j(v")$ with colour c"exists
            THEN
                v' = w
                ASSIGN the colour c' to the vertex v'
            ENDIF
        ENDWHILE
    ENDIF
ENDFOR
ENDPROC.
```

An example of the algorithm operation, for the colouring of the graph shown in Fig.5a, is presented in Tab.1. Only one recolouring phase takes place (step 6) and the colouring of the graph at this step is drawn in Fig.7.

Connection	Steps								
requests	0	1	2	3	4	5	6	7	8
(0,0)	A	A	A	A	A	A	A	A	A
(8,5)		A	A	A	A	A	A	A	A
(2,2)			B	B	B	B	B	B	B
(5,7)				A	A	A	A	A	A
(3,6)					C	C	B	B	B
(4,4)						B	C	C	C
(1,8)							C	C	C
(6,3)								B	B
(7,1)									C

B,C: Colours involved in the recolouring phase.

Tab.1 - Example of the Permutation Routing Algorithm operation for the colouring of the graph shown in fig.5a.

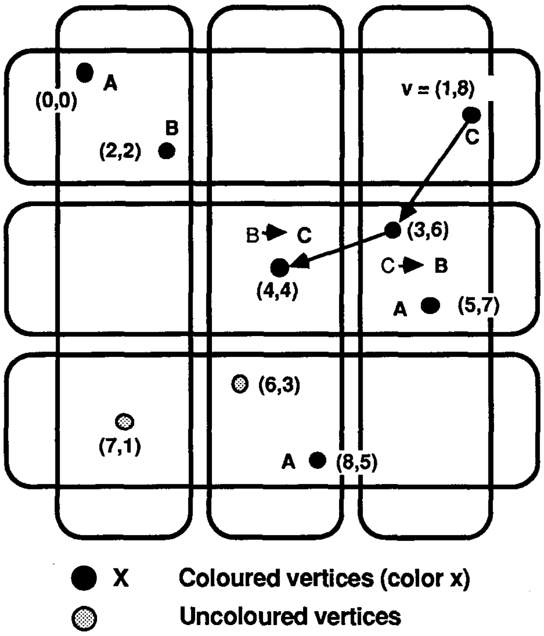


Fig.7 - Example of recolouring procedure.

The validation of the proposed algorithm is based on the following theorem:

Theorem 4: at the end of the step j (∀j), all the coloured vertices belonging to a clique of G^j have different colours.

Proof: the proof is inductive. At the end of the step j=0 only one vertex is coloured and the theorem is straightforwadly proved.

By hypothesis, at the beginning of the generic step j, chosen an uncoloured vertex v, since $\omega(G) = P$,the statement "COMPUTE..." always returns $AC[P, C^j_{in}(v)] \neq \emptyset$ and $AC[P, C^j_{out}(v)] \neq \emptyset$, i.e. each function provides a not empty set of colours for the vertex v. As a result, only two cases are possible: i) at least a colour c belonging to both these sets exists; ii) no colour belonging to both these sets exists.

In the case i) the colour c is assigned to v and the theorem is proved.

In the case ii) a recolouring phase takes place. This phase only concerns the subgraph $g^j(v,c',c")$ composed of the vertex v and all the vertices already labeled with the colours c' and c" chosen in the statement "SELECT...". In particular the recolouring procedure starts from the vertex v and follows a path[6] on $g^j(v,c',c")$.

In order to prove the theorem the following items must be verified:

a) the set of vertices belonging to $g^j(v,c',c")$ which can be reached from v constitutes a path on $g^j(v,c',c")$;

b) this path is not an odd cycle.

As far as the item a) is concerned, since at the end of the step j-1 the assertion of the theorem is supposed to be true, the subgraph $g^j(v,c',c")$ is perfect, its clique number equals 2 and its maximum degree is 2; so the item a) is verified.

As for the item b), it is trivially proved by considering that $g^j(v,c',c")$ is perfect and the clique number is 2. □

As far as the algorithm time complexity is concerned, though the general problem of graph colouring is NP–Complete, it can be seen that the complexity of the described Permutation Routing Algorithm is polynomial. It is worth noting that, during the step j, a colour is assigned to a new vertex and some vertices could be involved in a recolouring phase. By using a proper data structure, the complexity of the "COMPUTE..." and "IF a color.." statements, since the maximum number of available colors is P, is O(P). The recolouring process, since it follows a path which, in the worst case, is shorter than 2N/P steps, has a complexity equal to O(N/P)=O(P). Therefore the Permutation Routing Algorithm complexity is O(N·P) or, since $N=n^S$ and $P=n^{\lceil S/2 \rceil}$, $O(N^{3/2})$.

As a final remark, it is to be noted that the Permutation Routing Algorithm well suits to a telecommunication environment in which connection set-up and tear-down requests asynchronously occur. In fact, the algorithm does not require the whole permutation to be provided, but it has been designed to establish the connection requests accordingly their arrival process as well as to minimize the number of rerouting of the previously established connections. The achieved set-up time complexity of a single connection request is equal to O(P) or $O(N^{1/2})$.

6. Conclusions

In this paper the class of Parallel Delta Network (PDN) has been defined.

For such a class of networks the non-blocking conditions have been given. In particular it has been proved that the minimum number (L) of Delta subnetworks necessary to provide the rearrangeability property is $L=n^{\lceil S/2 \rceil -1}$. This result has been obtained by formulating the problem as a graph colouring one.

For the establishment of whatever inlet/outlet permutation, a routing algorithm has been proposed. It operates for any value of the SE size and shows a polynomial time complexity equal to $O(N^{3/2})$. However, it is worth noting that, as far as the set-up of a single connection request, this algorithm assures a time complexity equal to O(P) and a maximum number of rerouted paths lesser than 2N/P. This property makes it

[6] A path on a graph $\mathfrak{G}=[\mathcal{V},\mathfrak{E}]$ is sequence of vertices $w=\{v_1...v_k\}$, $k \geq 1$ such that $(v_j,v_{j+1}) \in \mathfrak{E}$ for $j=1,...,K-1$ without any repeated vertices.

well suitable to an asynchronous telecommunication environment.

ACKNOWLEDGEMENT

The authors wish to thank Prof. M. Lucertini for his fruitful suggestions and comments.

REFERENCES

[1] IEEE Computer Magazine Special Issue on "Interconnection Networks", Dec. 1983;

[2] C.Wu, T.Feng: "On a class of multistage interconnection networks", IEEE Trans. on Computers, vol. C-29, n°8, Aug. 1980, pp. 694–702;

[3] C.Wu, T.Feng: "The reverse-exchange interconnection networks", IEEE Trans. on Computers, vol. C–29, n°9, Sept. 1980, pp. 801–811;

[4] J.S.Turner: "New directions in communications or, which way to the Information Age", Proc. of 1986 Int. Zurich Seminar on Digital Communication, Zurich, Switzerland, March 11-13 1986;

[5] A. Huang, S. Knauer: "Starlite: a wideband digital switch", Proc. of Globecom 1984, Atlanta, U.S.A., Nov. 26-29 1984, pp. 121-125;

[6] C. Day, J.Giacopelli,J.Hickey: "Application of self–routing switches to lata fiber optic networks", Proc of I.S.S. 1987, Phoenix, U.S.A., March 15-20 1987, pp. 519-523;

[7] J.H.Patel: "Performance of processor-memory interconnection for multiprocessor", IEEE Trans. on Computers, vol. C-30, n°10, Oct. 1981, pp. 771–780;

[8] C.P.Kruskal, M.Snir: "The performance of multistage interconnection networks for multiprocessors", IEEE Trans. on Computers, vol. C-32, n12, Dec. 1983, pp. 1091-1099;

[9] V.Cherkassky, E.Opper, M.Malek: "Reliability and fault-diagnosis analysis of fault-tolerant multistage interconnection networks", Proc. of Int. Conference on Fault tolerant Computing System, Kissimmee, U.S.A., June 1984, pp. 246-251;

[10] J.P.Shen: "Fault tolerance analysis of several interconnection networks", Proc. of Int. Conference of Parallel Processing, 1982, pp. 102-112;

[11] V.E.Benes: "Mathematical theory of connecting networks and telephone traffic", Academic Press, New York, 1965;

[12] J.Lenfant: "Parallel permutations of data: a Benes network control algorithm for frequently used permuatations", IEEE Trans. on Computers, vol. C–27, n°7, July 1978, pp.637-647;

[13] G.F.Lev, N.Pippinger, L.G.Valiant: "A fast parallel algorithm for routing in permutation networks", IEEE Trans. on Computers, vol. C-30, n°2, Feb. 1981, pp. 93–100

[14] O.Nassimi, S.Sahni: "Parallel algorithms to set-up the Benes permutation networks", IEEE Trans. on Computers, vol. C-30, n°2, Feb. 1981, pp. 148-154;

[15] K.Y.Lee: "On the rearrangeability of 2·($\log_2$N)-1 stage permutation networks", IEEE Trans. on Computers, vol. C-34, n°5, May 1985, pp. 412–425;

[16] K.E.Batcher: "Sorting networks and their applications", Proc. of AFIPS Conference 1968, pp.307-314;

[17] M.Kumar, J.R.Jump: "Generalized Delta networks", Proc. of Int. Conference on Parallel Processing 1983, pp.10-18;

[18] S.M.Reddy, V.P.Kumar: "On fault tolerant multistage interconnection networks", Proc. of Int. Conference on Parallel Processing, 1984, pp. 155–164;

[19] D.P.Agrawal: "Graph theoretical analysis and design of multistage interconnection networks", IEEE Trans. on Computers, vol. C-32, n°7, July 1983, pp. 637-648;

[20] A.Varma, C.S.Raghavendra: "Realization of permutations on generalized Indra networks", Proc. of Int. Conference on Parallel Processing 1985, pp.328-333;

[21] M.Gondran, M.Minoux: "Graphs and algorithms", John Wiley, 1984;

[22] M.L. Golumbic: "Algorithmic graph theory and perfect graphs", Academic Press, New York 1980.