The C1-Networks: A Scalable Multistage Interconnection Network with Backward Links for Deflecting Routing

Zhen Chen, Zeng-Ji Liu and Zhi-Liang Qiu
National Key Lab of ISN, Xidian University
Xi’an, Shaan xi 710071, China.
E-mail: zhen@xidian.edu.cn

Abstract—In this paper, we propose the C1-Networks, including the Complete C1-Networks and the Incomplete C1-Networks, which are evolved from the B-Networks. The main difference between the Complete C1-Networks and the B-Networks is their connecting pattern between the stages. The available MINs to date are almost all restricted by the port number in the power of 2. The C1-Network is not under the control by this restriction, hence a good flexibility. In addition, the single-buffer performance model is developed and evaluated, and some results are given.

Keywords—Gamma network; B-network; C1-network; MIN; performance analysis; fault-tolerance

I. INTRODUCTION

Many multistage interconnection networks (MINs) have been proposed for multiprocessor systems and telecommunication network, such as Clos, Benes, Baseline, Omega, Indirect binary n-cube, etc. But MINs belonging to the data manipulator (DM) class [1][2][3][4] have been considered for this application only by a few researchers. The most well known network of the DM class is the Gamma Interconnection Network (GIN) [1][2]. To enhance the reliability of GIN, MGIN and CGIN [5][6] are proposed respectively.

The GIN is known to contain a cube network as a substructure. By preserving the cube structure but reversing the direction of all other links, the B-Network [7] is obtained from the GIN. In other words, B-Network is a MIN with backward links, if reverse direction of all the backward links in B-Network, a GIN is obtained. These backward links being used as alternate paths for requests blocked due to path or memory contentions, the performance of the B-Network [7][8] is greatly increased over that of the GIN with the same cost.

The important problem of building a MIN with arbitrary port number, somewhat surprisingly, has not been extensively addressed in the literature. The aforementioned MINs are all restricted by the power of two. For example, to preserve the cube structure (B0-Network), the B-Network also has the limitation that the port number of is the power of two. Thus, unfortunately the only choice for the architect is to design a MIN of the size equal to the closest power of two, which will make many switch elements unnecessary and wasted. One exception is [9], which handle the problem to design the binary shuffle-exchange network with any even number of ports.

In this paper, we focus on the data manipulator (DM) class, which shows the wraparound characteristic in connecting pattern between stages. We propose a new network, called C1-Network, which is evolved from the B-Network. Compared with B-Network, the C1-Network has different connecting pattern between stages. Also, we propose a routing algorithm, which make the network feasible and nearly as easy to use as when N is a power of two. Performance analysis shows that the Incomplete C1-Network behaves in a similar manner for power-of-two network sizes.

This paper is organized in the following way: In Section 2, the complete C1-Network (hereafter C1-Networks, for simplicity) is introduced in detail. The network configuration, the routing scheme and performance issues in C1-Network are also described. In Section 3 the Incomplete C1-Network is presented, and network configuration, routing scheme and performance issues are discussed correspondently. Some results are discussed in Section 4. Finally Section 5 concludes the paper.

II. THE C1-NETWORK

A. The C1-Network

Fig. 1. C1-Network. (A dash line represents a backward link).

A C1-Network of size \( N = 2^n \) consists of \( n+1 \) stages and each stage contains \( N \) switches. Every switch at intermediate stage is a \((3 \times 3)\) crossbar, while the two at the first and last stages are sizes \((2 \times 2)\). The stages are numbered from 0 to \( n \). The \( N \) sources are numbered from 0 to \( N-1 \) as well as the \( N \) switches at each stage from the top, and the \( N \) destinations. Connecting patterns between stages are based on the plus-minus- \( 2^n \) functions, namely,
the \( j \)-th switch at stage \( i \), \( 0 \leq i \leq n \), is connected to three switches at stage \( i+1 \) according to the three functions:
\[
f^*_j = j + (-2)^i \pmod{N}, \quad f^0_j = j, \quad f^m_j = j - (-2)^i \pmod{N},
\]
which respectively define the upper, straight, and lower connections originating from switch \( j \) at stage \( i \). The backward links are alternatively changed between the stages, i.e., the upper link, the lower link, then the upper link again and again. A \( 8 \times 8 \) C1-Network is depicted in Fig. 1.

As shown in Fig. 2, each \( 3 \times 3 \) SE has three input link, and three output links: upper input link, lower input link, and backward input link, and three output links: upper output link, lower output link, and backward output link.

\[\text{C1-Network stage 0.} \quad \text{C1-Network stage 1 (odd).} \quad \text{C1-Network stage i (even).} \quad \text{C1-Network stage n.}\]

\[\text{Fig. 2. The four types of switch in the C1-Networks.}\]

**B. Routing In The C1-Network**

A simple routing algorithm is based on the routing tag \( T \) which is a \((-2)\)-based binary representation of \((D - S) \mod N\), where \( D \) is the destination switch, \( S \) is the source switch, and \( N \) is the network size. The setting of a path from \( S \) to \( D \) corresponds to add the difference \((D - S)\) to \( S \) in modulo using connections available in the \( n \) stages.

For a path can only be represented in straight or \((-2)^i\) connections, the tag can be represented in the \( n-1 \)-bit \((-2)\)-based binary number:
\[
T = (t_{n-1} \ldots t_1 t_0)_{(-2)} = D - S \pmod{N}, \quad 0 \leq T \leq N - 1,
\]
In stage \( i \), if \( t_i = 0 \), the straight connection is used, if \( t_i = 1 \), the nonstraight connection is used.

Tag can be acquired as follows.

1. \((D - S) = T \pmod{N}, T \) can be a negative number.

2. Transforming the Tag \( T \) into \((-2)\)-based binary representation. It can be implemented by a division logic the same as a decimal number is transformed into the binary representation. The procedure in transformation the \(-9\) to a \((-2)\)-based binary representations is shown as the following example, \(-9\) is represented as \((1011)_{(-2)} \). Finally. The arrow is marked as the most significant bit (MSB).

The tag is thus acquired.

The performance analysis of the C1-Network is almost the same as the B-Network, because both networks have the same network topology and the same routing rule.

\[
\begin{array}{cccc}
-2 & 9 & \ldots & 1 \\
-2 & 5 & \ldots & 1 \\
-2 & 2 & \ldots & 0 \\
1 & \ldots & 1 & \uparrow
\end{array}
\]

**III. THE INCOMPLETE C1-NETWORK**

**A. The Definition of The Incomplete C1-Network**

An Incomplete C1-Network of size \( N' \) consists of \( n+1 \) stages and each stage contains \( N' \) switches, where \( 2^n < N' < 2^n \). Every switch at intermediate stage is a \((3 \times 3)\) crossbar, while the two at the first and last stages are sizes \((2 \times 2)\). The stages are numbered from 0 to \( n \). The \( N' \) sources are numbered from 0 to \( N'-1 \) as well as the \( N' \) switches at each stage from the top, and the \( N' \) destinations. Connecting patterns between stages are based on the plus-minus-2\(^i\) functions, namely, the \( j \)-th switch at stage \( i \), \( 0 \leq i \leq n \), is connected to three switches at stage \( i+1 \) according to the three functions:
\[
f^*_j = j + (-2)^i \pmod{N}, \quad f^0_j = j, \quad f^m_j = j - (-2)^i \pmod{N},
\]
which respectively define the upper, straight, and lower connections (upper output link, straight output link, and lower backward input link) originating from switch \( j \) at stage \( i \). The backward links are alternatively changed between the stages, i.e., the upper link, the lower link, then the upper link again and again. A \( 7 \times 7 \) C1-Network is depicted in Fig. 3.

**B. Routing In The Incomplete C1-Network**

The routing scheme in the Incomplete C1-Network is almost the same as the C1-Network, except that the rerouting rule is somewhat different.

The Incomplete C1-Network routing algorithm are also based on a routing tag \( T \) which is a representation of \((D - S) \pmod{N'}\) where \( D \) is the destination, \( S \) is the source, and \( N' \) is network size. The Tag can be written as:
\[
T = t_{n-1} \ldots t_0 = D - S \pmod{N'}, \quad 0 \leq T \leq N'-1.
\]

The arrow is marked as the most significant bit (MSB).
stage \( i \), if \( t_j = 0 \), the straight connection is used, if \( t_j = 1 \), the upper connection is used.

The backward links is also employed to provide alternate paths for rerouting the blocked packet at switches. The rerouting rule will be discussed in detail in the next section for performance modeling.

C. Performance Analysis of the single-buffer Incomplete C1-Network

1) Description

Notations:

(1) Let \( m_i (0 \leq i \leq n) \) denote the packet arrival rates on a straight forward output link of an SE in stage \( i \).

(2) Let \( m'_i (0 \leq i \leq n-1) \) denote the packet arrival rates on a nonstraight forward output link of an SE in stage \( i \).

(3) Let \( b_i (1 \leq i \leq n) \) as the packet arrival rate on a backward output link of an SE in stage \( i \).

(4) Let \( m \) denote the network input rate.

Rerouting Rules in Incomplete C1-Network:

The basic Rule is that a packet loses the contention will go back one stage and attends the contention at the next time. See Fig. 4 for details.

A packet is classified into two classes: the normal packet and the deflecting packet. A deflecting packet will change into a normal packet when it finish the rerouting and rejoin the contention where it failed at the last time.

(1) When the packet coming from two forward input at an SE request the straight output link at stage \( i \), where \( 1 \leq i \leq n-1 \). If it loses the contention with other packets (Every packet taking part in the contention has the equal probability to win), it is redirected to the backward output link.

(2) When the packet coming from two forward input links at an SE request the nonstraight output link at stage \( i \), where \( 1 \leq i \leq n-1 \). If it loses the contention with other packets (Every packet taking part in the contention has the equal probability to win), our scheme is just to discard it to make room for the fresh packet.

(3) All input packets from the backward links always request the nonstraight forward output link. When the packet coming from backward input link at an SE requests the nonstraight output link at stage \( i \), where \( 1 \leq i \leq n-1 \). If it loses the contention with other packets (Every packet taking part in the contention has the equal probability to win), Our scheme is just to discard it to make room for the fresh packet.

Assumptions:

For convenience, we assume that the desired destinations of the packets coming into a node at a given time are mutually independent. This assumption is intuitively reasonable and, without it, the analysis would be intractable. (It is an approximation because that the packets coming from a backward input link will always request the nonstraight link in the C1-network. For this assumption has been long used in the analysis of the B-network, thus we does not modify it).

We also assume that the traffic is distributed evenly over the forward output links. Using the routing tags appropriately for different source-destination pairs can do this. A routing tag selection strategy proposed in [10][11] can guarantee the even characteristic in links.

Now, we would consider the forward links and backward links separately. We also refer to the Fig. 2 in the following analysis.

Now, we would consider the forward links and backward links separately.

2) Forward Links

At Stage 0

At stage 0, an SE will have an external input link, a backward input link, and a pair of straight and nonstraight forward output links. As explained before, packets coming from a backward input link will always request a nonstraight output link. As a result, a straight output link of an SE in stage 0 will only carry packets from the external input link, whereas a nonstraight link carries packets both from the external input link and the backward input link. Therefore, we obtain:

\[
m_0 = \frac{m}{2}
\]

\[
m'_0 = 1 - \left(1 - \frac{m}{2}\right)(1 - b_1)
\]

At stage \( i \), where \( 1 \leq i \leq n-1 \)

For an SE in an intermediate stage, there are six links connected to it: a pair of straight and nonstraight forward input links, a backward input link, and a pair of straight and nonstraight forward output links. Similar to the case at stage 0, no packet coming from backward input link will be routed to the straight output link. Then, we obtain:

\[
m_j = 1 - \left(1 - \frac{m_{(i-1)}}{2}\right)\left(1 - \frac{m'_{(i-1)}}{2}\right)
\]
At Stage \( n \)

At the last stage, there is only a pair of straight and nonstraight forward input links to an SE. Therefore,

\[
m_n = 1 - (1-m_{(n-1)}) (1-m'_{(n-1)})
\]

3) Backward Links

At stage \( i \), where \( 1 \leq i \leq n-2 \)

For an SE at an intermediate stage \( i \), the backward output link will have a packet passing through only if two input packets of forward links are requesting for the straight forward output link, or one or more of them requests the nonstraight link with input packet of the backward input links.

If two of the input packets are requesting for the same nonstraight forward output link, only one can acquire this link, and the other will be lost in a C1-Network. (Realizing that all input packets from the backward input links always request the nonstraight forward output link.)

\[
h_i = \frac{1}{4} m_{(i-1)} m'_{(i-1)}
\]

At Stage \( n-1 \)

Note that the packet coming back from the backward input link will always request the backward output links, hence,

\[
h_{n-1} = h_n + \frac{1}{4} m_{n-2} m'_{n-2} (1-h_n)
\]

At Stage \( n \)

At the last stage, there will be a packet deflected back through the backward link only when both of the forward input links have incoming packets. That is,

\[
h_n = m_{n-1} m'_{n-1}
\]

4) Iterative Solutions of the Arrival Rates

We have derived the equations for the packet arrival rates in each link. Next, we need to find out \( m_n \), i.e., the normalized throughput. However, \( m_n \) cannot be solved recursively using the above equations alone. We have to solve it by tracing the behavior of the network for several successive stage cycles.

Now, let \( m_i \ (0 \leq i \leq n) \) and \( m'_i \ (0 \leq i \leq n) \) be, respectively, the packet arrival rates on a straight and nonstraight forward output link of an SE in stage \( i \) at stage cycle \( k \), and \( b_i \) as the packet arrival rate of a backward output link of an SE in stage \( i \) at stage cycle \( k \), \((0 \leq i \leq n)\). Then, rewriting the equations:

At Stage 0

\[
m'_i = 1 - \left( 1 - \frac{m_{(i-1)}}{2} \right) \left( 1 - \frac{m'_{(i-1)}}{2} \right) (1-h_{i+1})
\]

\[
m_k = \begin{cases} 0 & \text{if } k = 0 \\ \frac{m}{2} & \text{if } k \geq 1 \end{cases}
\]

\[
m_{i_0} = \begin{cases} 0 & \text{if } k = 0 \\ \frac{m}{2} & \text{if } k \geq 1 \end{cases}
\]

\[
m_k^{i_0} = \begin{cases} 0 & \text{if } k = 0 \\ \frac{m}{2} & \text{if } k \geq 1 \\ 1 - \left( 1 - \frac{m}{2} \right) (1-h_{i+1}) & \text{if } k \geq 2 
\end{cases}
\]

At stage \( i \), where \( 1 \leq i \leq n-2 \)

\[
m_k = \begin{cases} 0 & \text{if } k < i + 1 \\ 1 - \left( 1 - \frac{m_{(i-1)}}{2} \right) \left( 1 - \frac{m'_{(i-1)}}{2} \right) & \text{if } k \geq i + 1 \\
\end{cases}
\]

\[
m_k^{i} = \begin{cases} 0 & \text{if } k < i + 1 \\ 1 - \left( 1 - \frac{m_{(i-1)}}{2} \right) \left( 1 - \frac{m'_{(i-1)}}{2} \right) & \text{if } k \geq i + 1 \\
\end{cases}
\]

At Stage \( n-1 \)

\[
m_{n-1} = \begin{cases} 0 & \text{if } k < n \\ 1 - \left( 1 - \frac{m_{(n-2)}}{2} \right) \left( 1 - \frac{m'_{(n-2)}}{2} \right) & \text{if } k \geq n 
\end{cases}
\]

\[
b_k^{i} = \begin{cases} 0 & \text{if } k < n \\ \frac{1}{4} m_{(n-2)} m'_{(n-2)} & \text{if } k = n, n+1 \\ b_{n+1} (k-1) + \frac{1}{4} m_{(n-2)} m'_{(n-2)} (1-h_{i+1}) & \text{if } k \geq n + 2
\end{cases}
\]
1. \( m_{n-1}^k = \left\{ \begin{array}{ll}
0 & \text{if } k < n \\
1 - \left(1 - m_{n-2}^{k-1}\right) \left(1 - m_{n-2}^{k-1}\right) & \text{if } k = n, n+1 \\
1 - \left(1 - m_{n-2}^{k-1}\right) \left(1 - m_{n-2}^{k-1}\right) \left(1 - k_{n-1}^{k-1}\right) & \text{if } k \geq n + 2
\end{array} \right. 
\)

At stage \( n \)

\[ m_n^k = \left\{ \begin{array}{ll}
0 & \text{if } k < n + 1 \\
1 - \left(1 - m_{n-1}^{k-1}\right) \left(1 - m_{n-1}^{k-1}\right) & \text{if } k \geq n + 1 
\end{array} \right. 
\]

\[ b_n^k = \left\{ \begin{array}{ll}
0 & \text{if } k < n + 1 \\
m_{n-1}^{k-1} - m_{n-1}^{k-1} & \text{if } k \geq n + 1
\end{array} \right. 
\]

Having this set of equations, we can now solve \( m_n \) by tracing from \( k = 0 \). Some results are shown as follows in Fig 5(a) and Fig 5(b).

IV. CONCLUSIONS

We have proposed a new class networks, called the C1-Networks. The C1-Networks with size of a power of two have the same performance as the B-networks, while the Incomplete C1-Networks has little performance loss for the incompleteness. The performance of single buffer C1-Networks is developed and evaluated, and some results are given. The results show that the C1-Network has a good throughput with the backward links present. For the C1-Network is not under the control by the power of two, a C1-Network with an arbitrary port number can be built for practical use, hence a good flexibility.

ACKNOWLEDGMENT

This research was supported in part by the National Hi-Tech R&D Program under grant number 2002AA103062 and the Huawei Technologies Co. under contract number YBCN2002001.

REFERENCES