Class-Congruence Property and Two-Phase Routing of Borel Cayley Graphs

K. Wendy Tang and Bruce W. Arden

Abstract—Dense, symmetric graphs are useful interconnection models for multicomputer systems. Borel Cayley graphs, the densest degree-4 graphs for a range of diameters [1], are attractive candidates. However, the group-theoretic representation of these graphs makes the development of efficient routing algorithms difficult. In earlier reports, we showed that all degree-4 Borel Cayley graphs have generalized chordal ring (GCR) and chordal ring (CR) representations [2], [3]. In this paper, we present the class-congruence property and use this property to develop the two-phase routing algorithm for Borel Cayley graphs in a special GCR representation. The algorithm requires a small space complexity of $O(p + k)$ for $n = p \times k$ nodes. Although suboptimal, the algorithm finds paths with length bounded by $2D$, where D is the diameter. Furthermore, our computer implementation of the algorithm on networks with 1,081 and 15,657 nodes shows that the average path length is on the order of the diameter. The performance of the algorithm is compared with that of existing optimal and suboptimal algorithms.

Index Terms—**Generalized chordal ring, interconnection network, parallel computer.**

I. Introduction

A variety of symmetric graphs such as the *hypercube* and *toroidal mesh*, have been proposed as interconnection models [4], [5], [6], [7], [8], [9]. Systematic node labeling of these graphs can provide the bases for routing algorithms. In these systematic instances, labels of the source and destination node can be used to determine a next step which *optimally* reduces the *distance* to the destination. These *optimal, distance-reduction* routing schemes are easy to implement and thus making these graphs attractive. However, interconnection graph density becomes very important for massively parallel systems and unfortunately these systematically labeled graphs are not the densest graphs. (A *dense* graph has large number of nodes with a small diameter and degree. The *diameter* is the maximum of the minimal distance between all node pairs. The degree is the number of neighboring elements of a node.)

A special class of symmetric graphs, *Borel Cayley graphs* are currently, the densest known, degree-4 graphs for a range of diameters [1]. The definition of these graphs is reviewed in the next section. Originally, Borel Cayley graphs are defined over a group of matrices, which lack a simple ordering. Furthermore, connections are defined through modular matrix multiplication. In other words, routing or path determination between nonadjacent nodes is not trivial. The question arises whether ordering the nodes in some way and labeling them with integers can lead to an efficient routing algorithm, preferably based on a formula. That is, is there an *optimal, distance-reduction* formula based on node labels? None has been found for Borel Cayley graphs.

In this paper, we present the proof of the *class congruence property*, a property we discovered to be pertinent to a special representation of Borel Cayley graphs. Based on this property, we developed a *two-phase routing* algorithm that requires a small space complexity of $O(p + k)$ for $n = p \times k$ nodes. Although suboptimal, the algorithm finds paths with length bounded by $2D$, where D is the diameter of the network. Furthermore, our computer implementation of the algorithm on networks with 1,081 and 15,657 nodes shows that the average path length is on the order of the diameter. The performance of the algorithm is then compared with that of the existing algorithms.

This paper is organized as follows: In Section II, we review the definitions of GCR, CR, Cayley graphs and Borel Cayley graphs, and restate the propositions on the representations of the general Cayley and Borel Cayley graphs. An example from Borel Cayley graph is used to illustrate these representations. In Section III, we discuss the class-congruence property (CCP), a property pertinents to Borel Cayley graphs in a special GCR representation. Section IV presents the two-phase routing algorithm and use an example to illustrate the algorithm. Section V compares the performance of the algorithm with existing optimal and suboptimal algorithms. Finally in Section VI, we present a summary and conclusions.

II. Review

In this section we review the definitions of generalized chordal rings (GCR), chordal rings (CR) [9], [7], Cayley graphs [10], and Borel Cayley graphs [1].

A. GCR and CR Graphs

Definition 1. *A graph* $\mathbf{R}$ *is a generalized chordal ring (GCR) if nodes of* $\mathbf{R}$ *can be labeled with integers mod n, the number of nodes, and there is a divisor q of n such that node i is connected to node j iff node* $i + q$ *(mod n) is connected to node* $j + q$ *(mod n). A chordal ring (CR) is a special case of GCR, in which every node has +1 and −1 modulo n connections. In other words, a CR is a GCR and, in addition, all nodes on the circumference of the ring are connected to form a Hamiltonian cycle.*

B. Cayley and Borel Cayley Graphs

The construction of Cayley graphs is described by finite (algebraic) group theory. Recall that a group $(\mathbf{V}, *)$ consists of a set $\mathbf{V}$ which is closed under inversion and a single law of composition $*$, also known as group multiplication. There also exists an identity element $I \in \mathbf{V}$.

Definition 2. *A graph* $\mathbf{C} = (\mathbf{V}, \mathbf{G})$ *is a Cayley graph with vertex set* $\mathbf{V}$ *if two vertices* $v_1, v_2 \in \mathbf{V}$ *are adjacent* $\Leftrightarrow v_1 = v_2 * g$ *for some* $g \in \mathbf{G}$ *where* $(\mathbf{V}, *)$ *is a finite group and* $\mathbf{G} \subset \mathbf{V} \setminus \{I\}$. $\mathbf{G}$ *is called the generator set of the graph and* I *is the identity element of the finite group* $(\mathbf{V}, *)$.

The definition of a Cayley graph requires nodes to be elements in a group but does not specify a particular group. A *Borel Cayley graph* is a Cayley graph defined over the *Borel* subgroup of matrices:

Definition 3. *Let* $\mathbf{V}_{(p,a)}$ *be a set of Borel matrices, then*

$$\mathbf{V}_{(p,a)} = \left\{ \begin{pmatrix} x & y \\ 0 & 1 \end{pmatrix} : x = a^t \pmod p,\ y \in Z_p,\ t \in Z_k \right\}$$

where p and a are fix parameters, p is a prime, $a \in \mathbf{Z}_p \setminus \{0, 1\}$, *and k is the order of a. That is,* $a^k = 1$ *(mod p).*

In other words, the nodes of Borel Cayley graphs are 2×2 Borel matrices, and modular p matrix multiplication is chosen as the group operation $*$. Note that p and a are fixed parameters and the variables of a Borel matrix are $t \in \mathbf{Z}_k$ and $y \in \mathbf{Z}_p$. In other words, there are $n = |\mathbf{V}| = p \times k$ nodes. By choosing specific generators, Chudnovsky et al. [1] constructed the densest, nonrandom, degree-4 graph for diameter $D = 7, \ldots, 13$ from Borel Cayley graphs. It is also worth noting that the Borel Cayley graph discovered by Chudnovsky with

Manuscript received Mar. 16, 1992; revised February, 1993.
K.W. Tang is with the Department of Electrical Engineering, SUNY at Stony Brook, Stony Brook, NY 11794; e-mail: wtang@sbee.sunysb.edu.
B.W. Arden is with the Department of Electrical Engineering, University of Rochester, Rochester NY 14627.
To order reprints of this article, e-mail: transactions@computer.org, and reference IEEECS Log Number C95133.

$D = 11$ has $n = 38{,}764$. In our research, we have discovered yet another denser Borel Cayley graph with $n = 41{,}831$ for $D = 11$.

Borel Cayley graphs are defined over a group of matrices, which lack a simple ordering that is very helpful in the development of efficient routing schemes. Furthermore, in this original matrix definition, there is no concise description of connections. Adjacent nodes can be identified only through modular p matrix multiplications. The problem of finding an optimal path between nonadjacent nodes is not trivial. In earlier reports, we proved that all Borel Cayley graphs can be represented by GCR [2] and CR graphs [3]. These GCR/CR representations are useful for routing because nodes are defined in the integer domain and there is a systematic description of connections. We restate these propositions as follows:

PROPOSITION 1. *All finite Cayley graphs have GCR representations.*

PROPOSITION 2. *All degree-4 Borel Cayley graphs have CR representations.*

The proofs of these propositions are included in [2], [3] and therefore not repeated here. Basically, these representations are achieved by choosing a transform element $\mathbf{T}$ and class representing elements a_i, $i = 0, \ldots, q-1$ for the q classes of a GCR/CR from the vertex set. The choices of these elements are mainly arbitrary for the GCR case; and more specific for the CR case [2], [3].

C. An Example

As an example, consider a Borel Cayley graph with $p = 7$, $a = 2$, $k = 3$, and $n = 21$ nodes. For undirected, degree-4 Cayley graphs, the generator set $\boldsymbol{G} = \{\mathbf{A}, \mathbf{B}, \mathbf{A}^{-1}, \mathbf{B}^{-1}\}$. Suppose

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix},$$

the diameter $D = 3$.

To obtain a GCR representation, we arbitrarily choose the transform element

$$\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

with $\mathbf{T}^7 = \mathbf{I}$ and class-representing elements

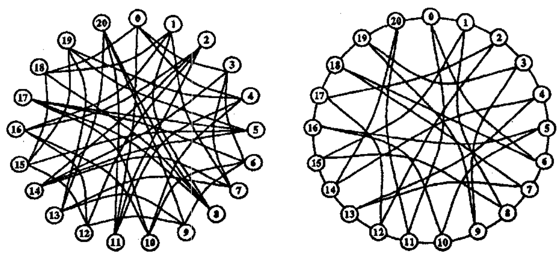
$$a_i = \begin{pmatrix} a^i & 0 \\ 0 & 1 \end{pmatrix}$$

for class i (see GCR algorithm in [2]). With these choices, the divisor $q = k = 3$ and $\mathbf{V} = \{0, 1, \ldots, 20\}$. Connections can be defined as: For any $j \in \mathbf{V}$, if j mod $3 = i$: j is connected to $j + \alpha_i, j + \alpha_i^{-1}, j + \beta_i$. and $j + \beta_i^{-1}$ (mod n), where the GCR constants $\alpha_i, \alpha_i^{-1}, \beta_i$ and β_i^{-1} are listed in Table I. This GCR representation is depicted in Fig. 1.

TABLE I
THE GCR/CR CONSTANTS

GCR Constants				
i	α_i	α_i^{-1}	β_i	β_i^{-1}
0	3	−3	4	−10
1	6	−6	7	−4
2	−9	9	10	−7

CR Constants							
i	0	1	2	3	4	5	6
γ_i	−10	7	10	6	9	5	−6
λ_i	6	−7	−6	−5	10	−10	−9



GCR Representation **CR Representation**

Fig. 1. A Borel Cayley graph $V_{(7,2)}$.

To obtain a CR representation, we choose the transform element $\mathbf{T} = \mathbf{A}^{-1}\mathbf{B}$ where $\mathbf{T}^3 = \mathbf{I}$ and class-representing elements $a_i = A^i$ for class i. With these choices, the divisor $q = 7$ and connections can be defined as: For any $j \in \mathbf{V}$, if j mod $7 = i$: j is connected to $j + 1, j - 1$, $j + \gamma_i$, and $j + \lambda_i$ (mod n), where the CR constants, γ_i and λ_i are listed in Table I. We show this CR representation of the graph in Fig. 1.

III. CLASS-CONGRUENCE PROPERTY (CCP)

In the transformation of a Cayley graph to a GCR, the choices of the transform element $\mathbf{T}$ and the class representing elements a_i are arbitrary (see GCR algorithm in [2]) By chosing specific choices of

$$\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

and

$$a_i = \begin{pmatrix} a^i & 0 \\ 0 & 1 \end{pmatrix}$$

for a Borel Cayley graph, we can provide a formulation of the GCR constants. This result is summarized in the following proposition:

PROPOSITION 3. *For any Borel Cayley graph with $n = p \times k$, and $a^k = 1$ (mod p), we assume the generators $\mathbf{A}$, $\mathbf{B}$ and their inverses to be:*

$$\mathbf{A} = \begin{pmatrix} a^{t_1} & y_1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} a^{t_2} & y_2 \\ 0 & 1 \end{pmatrix},$$

$$\mathbf{A}^{-1} = \begin{pmatrix} a^{k-t_1} & -a^{k-t_1} y_1 \\ 0 & 1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} a^{k-t_2} & -a^{k-t_2} y_2 \\ 0 & 1 \end{pmatrix}.$$

If we choose the transform element

$$\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

and the representing element of class i as

$$\begin{pmatrix} a^i & 0 \\ 0 & 1 \end{pmatrix},$$

we have a GCR representation of the graph with divisor $q = k$, and the graph is defined in GCR terms as: if vertex j belongs to class i, j is connected to $j + \alpha_i, j + \alpha_i^{-1}, j + \beta_i$, and $j + \beta_i^{-1}$ (mod n), corresponding to generators $\mathbf{A}$, $\mathbf{A}^{-1}$, $\mathbf{B}$, and $\boldsymbol{B}^{-1}$, where

$$\begin{aligned} \alpha_i &= < i + t_1 >_q + < a^i y_i >_p \quad q - i; \\ \alpha_i^{-1} &= < i - t_1 >_q + < -a^{i-t_1} y_1 >_p q - i; \\ \beta_i &= < i + t_2 >_q + < a^i y_2 >_p q - i; \\ \beta_i^{-1} &= < i - t_2 >_q + < -a^{i-t_2} y_2 >_p q - i; \end{aligned} \tag{1}$$

and $< i + t >_q$ denotes $(i + t)$ (mod q).

PROOF. By transforming the graph with

$$\mathbf{T}=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}$$

and choosing the representing element of class i as

$$\begin{pmatrix}a^i & 0\\ 0 & 1\end{pmatrix},$$

we partition the graph into $q = k$ classes:

$$\text{class } 0=\left\{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix},\ldots,\begin{pmatrix}1 & p-1\\ 0 & 1\end{pmatrix}\right\}$$

$$\text{class } 1=\left\{\begin{pmatrix}a & 0\\ 0 & 1\end{pmatrix},\begin{pmatrix}a & 1\\ 0 & 1\end{pmatrix},\ldots\begin{pmatrix}a & p-1\\ 0 & 1\end{pmatrix}\right\}$$

$$\vdots$$

$$\text{class } q-1=\left\{\begin{pmatrix}a^{q-1} & 0\\ 0 & 1\end{pmatrix},\begin{pmatrix}a^{q-1} & 1\\ 0 & 1\end{pmatrix},\ldots,\begin{pmatrix}a^{q-1} & p-1\\ 0 & 1\end{pmatrix}\right\}$$

According to this partition and the mapping defined in [2], matrix

$$\begin{pmatrix}a^i & y\\ 0 & 1\end{pmatrix}$$

is mapped to $i+yq$. The representing element of class i is

$$a_i=\begin{pmatrix}a^i & 0\\ 0 & 1\end{pmatrix},$$

and

$$a_i\mathbf{A}=\begin{pmatrix}a^{i+t_1} & a^i y_1\\ 0 & 1\end{pmatrix},\quad a_i\mathbf{A}^{-1}=\begin{pmatrix}a^{i-t_1} & -a^{i-t_1}y_1\\ 0 & 1\end{pmatrix}$$

$$a_i\mathbf{B}=\begin{pmatrix}a^{i+t_2} & a^i y_2\\ 0 & 1\end{pmatrix},\quad a_i\mathbf{B}^{-1}=\begin{pmatrix}a^{i-t_2} & -a^{i-t_2}y_2\\ 0 & 1\end{pmatrix}$$

According to the partition, these imply that i is connected to

$$<i+t_1>_q+<a^i y_1>_p q,\ <i-t_1>_q+<-a^{i-t_1}y_1>_p q,$$

$$<i+t_2>_q+<a^i y_2>_p q, \text{ and } <i-t_2>_q+<-a^{i-t_2}y_2>_p q$$

The formulae for $\alpha_i, \alpha_i^{-1}, \beta_i, \beta_i^{-1}$ thus follow. □

In other words, by choosing the transform element

$$\mathbf{T}=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix},$$

and the representing element of class i to be

$$\begin{pmatrix}a^i & 0\\ 0 & 1\end{pmatrix},$$

we impose a natural numbering system for the matrices in the group. It is this numbering system that allows us to deduce analytic formulae for the GCR constants, $\alpha_i, \beta_i, \alpha_i^{-1}$, and β_i^{-1} for class i. We notice that these constants are different for the different classes. However they are congruent modulo q. This implies that every class has the same class-connectivity and hence we name this property the *class-congruence property (CCP)*.

PROPOSITION 4. *The class-congruence property (CCP). The GCR constants associated with a Borel Cayley graph with*

$$\mathbf{T}=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix},$$

and

$$\begin{pmatrix}a^i & 0\\ 0 & 1\end{pmatrix},$$

are congruent modulo q, where q is the number of classes in the GCR. Specifically,

$$\begin{aligned} c_1 &= \alpha_i \pmod q = t_1 \quad \forall_i;\\ c_2 &= \alpha_i^{-1} \pmod q = q-t_1\ \forall_i;\\ c_3 &= \beta_i \pmod q = t_2 \quad \forall_i;\\ c_4 &= \beta_i^{-1} \pmod q = q-t_2\ \forall_i. \end{aligned} \tag{2}$$

The proof of this proposition is simply the modulo q arithmetic of the connection constants $\alpha_i, \alpha_i^{-1}, \beta_i$, and β_i^{-1} in (1).

We can verify (1) and (2) with the 21-node Borel Cayley graph described in Section II.C. In that section, we have a GCR representation of the graph with choices of $\mathbf{T}$ and a_i that match the specifications in Propositions 3 and 4. A simple substitution shows that the values of $\alpha_i, \alpha_i^{-1}, \beta_i$, and β_i^{-1} in Table I satisfy (1) and (2). In other words, even though the GCR constants, $\alpha_i, \alpha_i^{-1}, \beta_i$, and β_i^{-1} are different for the three classes, they are congruent modulo $q = 3$. This property is useful for routing because it facilitates the decoupling of the original graph into two smaller subgraphs. We call the resulting algorithm the *two-phase routing* algorithm.

IV. TWO-PHASE ROUTING

In general, the goal of *routing* is to send messages between pairs of nodes. There are two aspects: *path identification* between nonadjacent nodes, and how to resolve *conflicts* when multiple messages in a node have the same outgoing links. In this paper, we discuss the first aspect: path identification.

Path identification is a trivial problem for graphs with *path-defining* labels that implicitly define shortest paths between vertices. In this case, *optimal routing* or *shortest-path identification* can be achieved computationally with an algorithm that has a space requirement independent of graph size, i.e., its space complexity is O(1). The toroidal mesh [11] and hypercube [6] are examples of such graphs. However, there is no existing path-defining label for Borel Cayley graphs.

When Borel Cayley graphs are represented in the special GCR representation specified in Proposition 3, two optimal routing (path identification) schemes become feasible. The first algorithm is a progressive, table look-up scheme that can be applied to any network in the integer domain [12]. The second algorithm, called *vertex transitive routing*, is based on the symmetric property of the network [12]. Both algorithms are *optimal* in the sense that shortest paths are guaranteed. However, both have a large space commitment of $O(n)$ at each node, or $O(n^2)$ for the entire network of size n. To reduce the space-commitment, we propose a *suboptimal* routing algorithm the *two-phase routing* for Borel Cayley graphs in the special GCR representation specified in Proposition 3. The algorithm is suboptimal only because shortest paths are not guaranteed. The performance of the algorithm is evaluated in Section V.

We call this algorithm "two-phase" routing because the original large Borel Cayley graph with $n = p \times k$ nodes is divided into two smaller graphs. Such decoupling is made simpler because of the class-congruence property of Borel Cayley graphs described in Section III. The algorithm is divided into two phases. In Phase I, we have a degree-4 GCR graph of size $n_1 = k$. In Phase II, we have another graph with size $n_2 = p$. In essence, Phase I deals with class-to-class routing while Phase II routes messages within the same class. A message is first sent to an arbitrary node of the same class as the final destination; then in Phase II, it is routed to the destination node. We describe these two phases separately:

A. Phase I: Class-to-Class Routing

This phase of the algorithm is responsible for routing messages from the source node to an arbitrary node of the destination class. We can, therefore, consider a smaller graph with size $n_1 = k$ for the $q = k$ original classes. Through the use of the class-congruence property, we proceed to show that this smaller graph is actually a GCR with one class.

Proposition 3 provides explicit formulae for the connection constants $\alpha_i, \alpha_i^{-1}, \beta_i, \beta_i^{-1}$ for class i of a Borel Cayley graph in a special GCR representation. Furthermore, Proposition 4 (the class-congruence property) shows that:

$$\alpha_i = c_i = t_i,\ \alpha_i^{-1} = c_2 = -t_1,\ \beta_i = c_3 = t_2,\ \beta_i^{-1} = c_4 = -t_2 (\text{mod } q)\ \forall_i.$$

The fact that c_1 to c_4 are independent of class implies the "class connection" pattern of the original graph is the same for nodes in different classes. For any node j in class i, j is connected to nodes in class $i + t_1$, $i - t_1$, $i + t_2$, $i - t_2$ (mod q) through generators $\mathbf{A}$, $\mathbf{A}^{-1}$, $\mathbf{B}$, and $\mathbf{B}^{-1}$, respectively. That is, each node in Phase I which actually represents a class in the original graph has the same connectivity constants, c_1 to c_4. In other words, we have a smaller and simpler GCR graph with k nodes and one class. We can, therefore, apply the vertex-transitive routing algorithm (see [12] for details) for this smaller GCR graph. With only k nodes, the space complexity for this phase is reduced to $k \times \delta$ and the time complexity is $O(D_1)$, where D_1 is the diameter for this phase, $D_1 \le D$, and δ is the degree of the graph. With only one class, vertex transitivity becomes simple: If i connects to j by a sequence of generators, 0 connects to $j' = j - i$ through the same sequence of generators.

Besides using the vertex-transitive algorithm in this phase, the fact that this phase involves a simple GCR with just one class facilitates an entirely computational algorithm without any storage requirement. Such an algorithm is described as follows:

A.1 A Computational Phase I Algorithm

From Proposition 4, we assume a GCR graph with k nodes and just one class. That is, for any vertex i in this phase, i is connected to c_1, c_2, c_3, and c_4, where

$$c_1 = t_1, \quad c_2 = -t_1, \quad c_3 = t_2, \quad c_4 = -t_2.$$

In other words, we have a simple GCR with k nodes and the constants are: $\pm t_1$, $\pm t_2$, and diameter is D_1. Without lost of generality, we assume $t_1 > t_2$. Then for any node $r \in \mathbf{V}$.

$$r = m_1 t_1 + m_2 t_2 \quad (\text{mod } k)$$

where $-D_1 \le m_1, m_2 \le D_1$ and $|m_1| + |m_2| \le D_1$. The problem of routing is the same as identifying m_1 and m_2. We define the following constants l_1, l_2, and Q such that

$$l_1 = \frac{D_1 t_1}{k},\ l_2 = \frac{D_1 t_2}{k},\ Q = \frac{D_1 t_2}{t_1}$$

Then

$$r = m_1 t_1 + m_2 t_2 \quad (\text{mod } k)$$

$$\Rightarrow \quad m_1 t_1 + m_2 t_2 = \begin{cases} r + l\,k & \text{or} \\ r - (l+1)\,k & \end{cases}$$

where $0 \le l \le l_1 + l_2$. Consider

$$\begin{aligned} r + l\,k &= m_1 t_1 + m_2 t_2 \\ &= (m_q \pm Q')t_1 \pm d \qquad 0 \le Q' \le Q, |m_2 t_2| = Q' t_1 + d \\ \Rightarrow (r + l\,k)/t_1 &= m_1 + Q' \qquad -Q - 1 \le Q' \le Q \\ \Rightarrow m_1 &= (r + lk)/t_1 - Q' \\ m_2 &= \frac{(r + lk) - m_1 t_1}{t_2} \end{aligned}$$

Similarly,

$$\begin{aligned} r - (l+1)\,k &= m_1 t_1 + m_2 t_2 \\ \Rightarrow -m_1 &= \frac{(l+1)k - r}{t_1} - Q' \qquad -Q - q \le Q' \le Q \\ -m_2 &= \frac{(l+1)k - r + m_1 t_1}{t_2} \end{aligned}$$

We summarize this iterative algorithm in Table II. There is no storage space required for this algorithm. However, the time complexity is of $O(D_1^2)$. Once a message reached a node in the same class as the final destination, either by the vertex-transitive algorithm or by the computational scheme, we proceed to the second phase.

TABLE II
A COMPUTATIONAL PHASE I ALGORITHM

```
for (l = 0; l ≤ l1 + l2; ++ l)
for (Q' = −Q − q; Q' ≤ Q; ++Q)
{       m1 = (r + l k) / t1 − Q';
        if (r + l k − m1t1) mod t2 = = 0
        {  m2 = (r + l k − m1t1)/ t2;
           print (m1, m2)}
        m1 = ((l + 1)k − r) / t1 − Q';
        if ((l + 1) k − r + m1t1) mod t2 = = 0
        {  m2 = ((l + 1) k − r + m1t1)/t2;
}       print (−m1, −m2)}
```

B. Phase II: Within Class Routing

In this phase, we consider routing messages within the destination class. We assume both source and destination nodes belong to the same class. In other words, the source node $i = m_1 q + c_1$ and the destination node $j = m_2 q + c_2$, where $c_1 = c_2$. Our original vertex-transitive property (see [12] for details) becomes: if i connects to j through a sequence of generators, then vertex 0 connects to vertex j' through the same sequence of generators, where

$$\begin{aligned} j' &= < a^{k-c_1}(m_2 - m_1) >_p q + < c_2 - c_1 >_q \\ &= < a^{k-c_1(m_2-m_1)} >_p q, \end{aligned} \tag{3}$$

where $< x >_p$ denotes x (mod p). That is, if i and j are different by some multiple of q, j' is also some multiple of q. Because of this property, we can establish a database that stores all the paths from node 0 to nodes $q, 2q, \ldots, (p-1)q$. Routing between any nodes i and j can be achieved by looking up the corresponding row j' of the table. To conclude, this phase needs a table of O(p) and the time complexity is of $O(D_2)$ where p is the number of nodes and $D_2 \le D$ is the diameter in this phase. The following example illustrates this algorithm.

C. An Example

In this example, we find a path between vertex 0 and 16 of the Borel Cayley graph in GCR representation described in Section II.C. This

Borel Cayley graph has $p = 7$, $n = 21$ nodes, $q = 3$ classes, diameter $D = 3$, and the connectivity are defined as: For any $i \in \mathbf{V}$, if i mod 3 =:

$$\begin{aligned} &"j": \text{i is connected to } i+\alpha_j, i+\alpha_j^{-1}, i+\beta_j, i+\beta_j^{-1}; \\ &"0": \text{i is connected to } i+3, i-3, i+4, i-10; \\ &"1": \text{i is connected to } i+6, i-6, i+7, i-4; \\ &"2": \text{i is connected to } i-9, i+9, i+10, i-7. \end{aligned} \tag{4}$$

Furthermore,

$$\begin{aligned} c_1 &= \alpha_j \pmod q = 0 \ \forall_j; \\ c_2 &= \alpha_j^{-1} \pmod q = 0 \ \forall_j; \\ c_3 &= \beta_j \pmod q = 1 \ \forall_j; \\ c_4 &= \beta_j^{-1} \pmod q = -1 \ \forall_j. \end{aligned}$$

In Phase I, we have a simple GCR with three nodes where each one connects to ±1. In other words, routing can be achieved in a single step; taking path **B** for +1 and taking path $\mathbf{B}^{-1}$ for −1. In this example, source node i = class (0) = 0, destination node j = class (16) = 1. We apply our vertex-transitive formula with $j' = < j - i >_q = 1$. That is, taking path **B** gets to the correct destination class. Furthermore our GCR constants show that taking path **B** corresponds to +4 in class 0. Hence our problem now becomes finding a path between node 4 and node 16, both of which belong to class 1.

In Phase II, we apply our vertex-transitive algorithm to a graph with p nodes. Accordingly, we have a table of size $(p - 1)D$. Such a table is shown in Fig. 2. We apply our vertex-transitive formula (3) for this phase and find $j' = 2q$. We then look up row 2 of the database and find the corresponding path to be: **AA**. This concludes the routing process and we have found path **BAA** between nodes 0 and 16.

q	**A**		
2 q	**A**	**A**	
3 q	**A**	**A**	**A**
4 q	$\mathbf{A}^{-1}$	$\mathbf{A}^{-1}$	$\mathbf{A}^{-1}$
5 q	$\mathbf{A}^{-1}$	$\mathbf{A}^{-1}$	
6 q	$\mathbf{A}^{-1}$		

Fig. 2. A phase II routing table for $\mathbf{BL}_2(\mathbf{Z}_7)$.

V. Performance Evaluation

We use a computer program to implement the two-phase routing algorithm. To evaluate the performance of the algorithm, a message is sent from an arbitrary source node, say node 0, to all other nodes in the network. The path length obtained through two-phase routing is recorded and compared with the optimal (shortest path) case.

The performance of the algorithm is also compared with another suboptimal routing algorithm, *CR routing*. CR routing exploits the CR representations of Borel Cayley graphs. For this algorithm, each node stores only two CR constants in addition to an implied +1 and −1. Messages are routed to intermediate nodes that decrease the *peripheral distance* from the destination node. Here *peripheral distance* refers to distance around the circumference of the CR graph. For example, node 1 is closer to node 3 than to node 4 in the *peripheral* sense. For obvious reasons, paths obtained by this algorithm are suboptimal in length. Instead of choosing an intermediate node from the immediate neighbor of the source node, a more dynamic approach is to choose intermediate nodes from all nodes within a certain distance d from the source. In other words, the source node "looks ahead" a certain distance and routes the message towards the node that is "closest" (in the peripheral sense) to the destination node. This dynamic approach requires more storage, $2q = 2k$ constants, instead of two constants, need to be stored in each node.

We investigate large Borel Cayley graphs with two different values of p. The first case deals with $p = 47$, $k = 23$, $a = 2$, and $n = 1{,}081$, and in the second case $p = 307$, $k = 51$, $a = 4$, and $n = 15{,}657$. In both cases, we consider graphs with four different sets of generators and hence different diameters, the first of which corresponds to the densest degree-4 graphs. We assume the following notations: t_1 and t_2 define the generators:

$$\mathbf{A} = \begin{pmatrix} a^{t_1} & 1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} a^{t_2} & 1 \\ 0 & 1 \end{pmatrix},$$

D stands for the diameter, $\overline{avg}$ is the deterministic average path length, which is determined by taking the average of all optimal path lengths between any two nodes. The following parameters are obtained from the program: *avg* for the average path length, and *max* for the maximum path length. The results are summarized in Table III and Table IV. In the case of CR routing, parameter d corresponds to the different "look ahead" distance. For $d = 1$, only the immediate neighbors are considered and for $d = 4$, the neighbors at distance 1, …, 4 are considered. Furthermore, the average path length and the path length distribution for the densest cases are also plotted in Figs. 3 and 4.

From these results, we observe that the two-phase routing has average path length comparable with the diameter D and maximum path length bounded by $2D$. The CR routing, on the other hand, has large path lengths in general, even though its performance improves with the "look ahead" distance d at the expense of a higher time complexity. To conclude, two-phase routing gives good performance (maximum path lengths are bounded by $2D$ and average is close to the diameter D) with a small space complexity of $O(p + k)$, where $n = p \times k$.

TABLE III
Simulation Results for $p = 47$, $n = 1{,}081$

$p = 47, k = 23, a = 2, n = 1{,}081$		set 1	set 2	set 3	set 4
t_1		1	7	1	3
t_2		7	8	2	6
D		7	8	8	9
$\overline{avg}$		5.54	5.74	5.76	5.72
D_1		4	6	6	6
D_2		7	7	7	7
CR Routing	$d = 1$ max	50	39	50	41
	avg	24.50	18.40	20.85	17.64
	$d = 2$ max	35	32	37	33
	avg	13.67	14.17	14.24	12.68
	$d = 3$ max	25	26	24	20
	avg	9.71	10.83	10.16	9.31
	$d = 4$ max	16	17	22	18
	avg	6.65	7.33	7.67	7.53
Two-Phase Routing	max	11	13	13	13
	avg	7.67	8.12	8.50	8.03

TABLE IV
SIMULATION RESULTS FOR $p = 307$, $n = 15{,}657$

$p = 307, k = 51, a = 4, n = 15{,}657$		set 1	set 2	set 3	set 4
t_1		2	1	4	1
t_2		16	4	13	2
D		10	11	12	15
$\overline{avg}$		8.10	8.16	8.56	9.65
D_1		6	7	8	13
D_2		10	9	10	10
CR Routing	$d = 1$ max	125	240	152	245
	avg	52.76	130.63	69.81	120.53
	$d = 2$ max	115	112	132	227
	avg	49.99	62.65	51.24	121.42
	$d = 3$ max	100	76	107	91
	avg	35.56	31.04	38.88	32.31
	$d = 4$ max	55	51	68	75
	avg	20.64	21.82	24.30	24.69
Two-Phase Routing	max	16	16	18	23
	avg	11.49	11.38	12.37	13.99

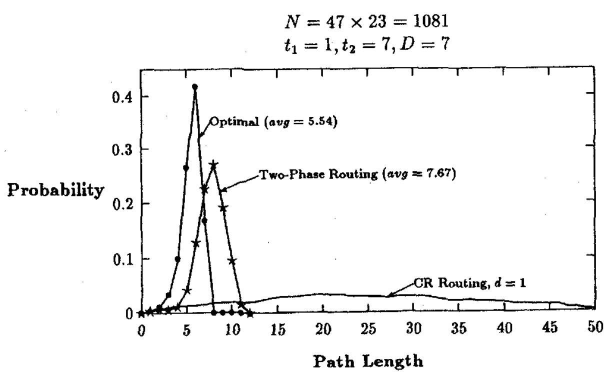


Fig. 3. Path length distribution.

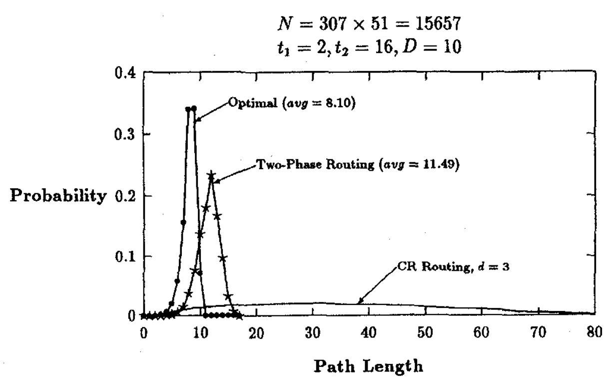


Fig. 4. Path length distribution.

VI. CONCLUSIONS

A variety of network topologies have been proposed as efficient interconnection networks. In many cases, the networks are symmetric and have systematic vertex labels. Furthermore, knowing the vertex labels of the source and destination often permits the *optimal* choices of the next step in a multistep path. These choices are *optimal* in the sense that the distance to the destination node is reduced. Such *distance-reduction* routing property is essential in the efficient implementation of the network. However, these systematically labeled graphs are not the densest, an important factor in the construction of massively parallel systems.

A special class of symmetric graphs, *Cayley graphs*, has received special attention as interconnection models [1], [13], [14]. One of its subclass, *Borel Cayley graphs*, are currently the densest known, constructive, degree-4 graphs for diameter $D = 7, \ldots, 13$ [1]. These Borel Cayley graphs are originally defined over a group of matrices, the Borel matrices. Unlike other existing networks, Borel Cayley graphs lack a systematic vertex labeling that can induce a distance-reduction routing algorithm. For Borel Cayley graphs, knowing the labels of the source and destination nodes does not render the determination of a path. In other words, path determination between nonadjacent nodes is a not a trivial problem.

In earlier research effort, we have proved that all Borel Cayley graphs have *GCR* and *CR* representations. These GCR/CR graphs are existing topologies, defined in the integer domain. Furthermore, there is a concise description of connections based on *class-structure*. This novel concept of transforming Borel Cayley graphs into GCR/CR made two optimal routing schemes, a *general* table look-up scheme and the *vertex-transitive routing*, feasible [15], [12]. However, these schemes require a space commitment of $O(n)$ at every node and $O(n^2)$ for the entire network of n nodes.

In this paper, we presented and proved a property pertinent to Borel Cayley graphs in a special GCR representation, the *class-congruence property (CCP)*. Based on this property, we developed a suboptimal routing algorithm with a smaller space complexity, the *two-phase routing*. Its space commitment is of $O(p + k)$ with time complexity $O(D)$, or $O(p)$ with time complexity $O(D^2)$. Even shortest paths are not guaranteed, the path length is bounded by $2D$, where D is the diameter. Computer implementation of the algorithm shows that the average path length is close to the diameter. The performance of the algorithm is also compared with another suboptimal routing algorithm, the CR routing. The results indicated that two-phase routing gives much shorter path lengths than that of CR routing.

REFERENCES

[1] D.V. Chudnovsky, G.V. Chudnovsky, and M.M. Denneau, "Regular graphs with small diameter as models for interconnection networks," Tech. Report RC 13484(60281), IBM Research Division, Feb. 1988.

[2] B.W. Arden and K.W. Tang, "Representation and routing of Cayley graphs," *IEEE Trans. Comm.*, vol. 39, no. 11, pp. 1,533-1,537, Nov. 1991.

[3] K.W. Tang and B.W. Arden, "Representations for Borel Cayley graphs," *SIAM J. on Descrete Math.* vol. 6, no. 4. pp. 655-676, Nov. 1993.

[4] T.Y. Feng, "A survey of interconnection networks," *Computer*, vol. 14, pp. 12-27, Dec. 1981.

[5] G.H. Barnes, "The ILLIAC IV computer," *IEEE Trans. Computers*, vol. 17, no. 8, pp. 746-757, Aug. 1968.

[6] J.P. Hayes, "Architecture of a hypercube supercomputer," *IEEE Proc. 1986 Int'l Conf. Parallel Processing*, pp. 653-660, 1986.

[7] B.W. Arden and H. Lee, "Analysis of Chordal ring network," *IEEE Trans. Computers*, vol. 30, no. 4, pp. 291-295, Apr. 1981.

[8] F.P. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Comm. ACM*, pp. 300-309, May 1981.

[9] J.C. Bermond and C. Delorme, "Strategies for interconnection networks: Some methods from graph theory," *J. Parallel and Distributed Computing*, vol. 3, pp. 433-449, 1986.

[10] N. Biggs, *Algebraic Graph Theory*. London: Cambridge Univ. Press, 1974.

[11] M. Ben-Ayed, "Dynamic routing for regular direct computer networks," PhD thesis, Electrical Engineering Dept., College of Engineering and Applied Science, Univ. of Rochester, New York, 1990.

[12] K.W. Tang and B.W. Arden, "Vertex-transitivity and routing for Cayley graphs in GCR representations," *Proc. 1992 Symp. Applied Computing*, Kansas City, Mo, pp. 1,180, 1,187, Mar. 1992.
[13] S.B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 555-565, Apr. 1989.
[14] G.E. Carlsson, J.E. Cruthirds, and H.B. Sexton, "Interconnection networks based on a generalization of cube-connected cycles," *IEEE Trans. Computers*, vol. 334, no. 8, pp. 769-772, Aug. 1985.
[15] D.A. Reed and R.M. Fujimoto, *Multicomputer Networks*. Cambridge, Mass.: MIT Press, 1987.

Survival Reliability of Some Double-Loop Networks and Chordal Rings

Frank K. Hwang and Paul E. Wright

***Abstract*—Three of the most well known double-loop networks are the *distributed double-loop computer network* (DDLCN), the *daisy chain*, and the braided ring. We compute the exact reliabilities of all three networks. We also extend the results on double-loop networks to directed chordal rings.**

***Index Terms*—Double-loop network, daisy chain, fault tolerance, reliability, consecutive-2-out-of-*n* system, chordal ring.**

I. Introduction

Distributed loop architectures have become increasingly popular in recent years for local area networks due to their low cost and high performance. However, a simple unidirectional loop is unreliable since any failure in an interface or communication link destroys the function of the loop. On the other hand a multiloop requires complicated control and routing at the interfaces. Therefore the double loop is a happy compromise. The three most well-known double-loop networks are the distributed double-loop computer network (DDLCN) of Liu et al. [11], [12], [21], [22] which is used in the FDDI network [18], the daisy chain of Grnarov, Kleinrock, and Gerla [3], and the braided ring which is used in SILK [7]. Another way to add reliability and performance to a simple loop is to add a single link to each interface. The bidirectional version was first proposed by Arden and Lee [1] and the network is called a *chordal ring*. Here we considered the unidirectional version called a *directed chordal ring*.

Two measures of reliability have often been used. The *terminal–pair reliability* [3], [5], [14], [15], [16], [17] averages the 2–terminal reliability over all ordered pairs of nodes. The *survival reliability* [2], [6], [7], [8], [9], [19], [20] computes the probability that the set of working (surviving) nodes is strongly connected. Computing either the terminal-pair reliability or the survival reliability is a difficult problem. Even though node failures and link failures are usually not treated together except in [3], [13], exact reliabilities were given only for the DDLCN in the terminal-pair reliability case [3], [13] and for the braided ring in the survival reliability case [2], [8]. Exact reliabilities were also given under additional conditions such a as small number of nodes or failed nodes [5], [14], [15], [16], [19] or the existence of a central-control node and every 2-terminal connection is from or to this node [10]. Approximations and bounds were given elsewhere [6], [9], [19]. In particular, Grnarov, Kleinrock, and Gerla [3] assumed that both nodes and links can fail with distinct probabilities and gave approximate terminal-pair reliability for the daisy chain.

In this paper, we not only allow failures of both nodes and links, we also allow links of the two loops to fail with different probabilities to accommodate the case that the two types of links have different lengths or constructions (the approach works for the more general case that each node and link has its own failure probability). We present a new approach and the first efficient method to compute exact survival reliabilities for both the daisy chain and the braided ring (the DDLCN is easy). We also prove a relation between directed chordal rings and a class of double loop networks (which include the three above-mentioned networks) such that any reliability results for the latter applies to the former.

II. A General Approach

A double-loop network $DL(n, a, b)$ can be represented by a digraph with n nodes $0, 1, \ldots, n-1$ and $2n$ links, n of each type: $i \to i + a \bmod n$, $i \to i + b \bmod n$, $i = 0, 1, \ldots, n - 1$. Thus, $DL(n, 1, n-1)$ is the DDLCN, $DL(n, 1, n-2)$ is the daisy chain and $DL(n, 1, 2)$ is the braided ring (see Fig. 1). We assume that each node has failure probability q $(= 1 - p)$, each a-link fails with probability q_a $(= 1 - p_a)$, each b-link fails with probability $q_b = (1 - p_b)$, and the states of all nodes and links are independent. Let $R(n, a, b)$ denote the survival reliability of $DL(n, a, b)$. For completeness, we define $R(1, a, b) = 1$.

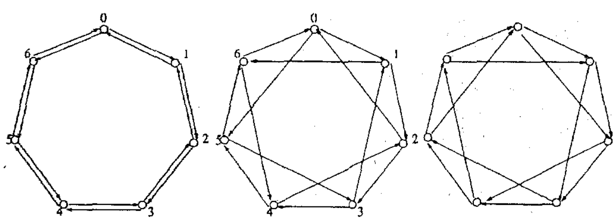


Fig. 1. (a) $DL(7, 1, 6)$; (b) $DL(7, 1, 5)$; (c) $DL(7, 1, 2)$.

THEOREM 2.1. *For* $n \geq 2$,

$$R(n, 1, n-1) = n\sum_{k=1}^{n-2} q^k p^{n-k} (p_a p_b)^{n-k-1} + p^n \left\{ \left[p_a^n + p_b^n - (p_a p_b)^n \right] + n q_a q_b (p_a p_b)^{n-1} \right\}. \tag{2.1}$$

PROOF. It is easily seen that a DDLCN fails if there exist two nonadjacent failed nodes. Suppose that there exists a sequence of k, $1 \leq k \leq n-2$ consecutive failed nodes. Then the $n-k-1$ a-links and the $n-k-1$ b-links between the surviving $n-k$ nodes must all work for the two end nodes to communicate. Since there are n ways of selecting k consecutive failed nodes, each summand in the first term of (2.1) captures the probability of such an event for a fixed k.

If all nodes work, then the network fails if there exist one failed a-link and one failed b-link which are not coupled (i.e., one is $i \to i + 1$ and the other is $i + 1 \to i$). The first member of the second term of (2.1) gives the probability of the event that either all a-links or all b-links work, while the second member accounts for the event that the (single) pair of failed nodes is coupled. □

Manuscript received Nov. 30, 1993; revised May 17, 1994.
The authors are with AT&T Bell Laboratories, Murray Hills, NJ 07974; e-mail: fkh@research.att.com
To order reprints of this article, e-mail: transactions@computer.org, and reference IEEECS Log Number C95142.