Computer Science
Technical Report

Colorado State University

Transposition Networks as a Class of Fault-Tolerant Robust Networks

Shahram Latifi
Department of Electrical Engineering
University of Nevada
Las Vegas, NV 89154

Pradip K. Srimani
Department of Computer Science
Colorado State University
Ft. Collins, CO 80523

Technical Report CS-95-104

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (303) 491-5792 Fax: (303) 491-2466
WWW: http://www.cs.colostate.edu

Transposition Networks as a Class of Fault-Tolerant Robust Networks

Shahram Latifi
Department of Electrical Engineering
University of Nevada
Las Vegas, NV 89154

Pradip K. Srimani
Department of Computer Science
Colorado State University
Ft. Collins, CO 80523

Abstract

The paper proposes designs of interconnection networks (graphs) which can tolerate link failures. The networks under study belong to a subclass of Cayley graphs whose generators are subsets of all possible transpositions. We specifically focus on star and bubble-sort networks. Our approach is to augment existing dimensions (or generators) with one or more dimensions. If the added dimension is capable of replacing any arbitrary failed dimension, it is called a wildcard dimension. It is shown that, up to isomorphism among digits used in labeling the vertices, the generators of the star graph are unique. The minimum number of extra dimensions needed to acquire i wildcard dimensions is derived for the star and bubble-sort networks. Interestingly, the optimally augmented star network coincides with the Transposition network, T_n. Transposition networks are studied rigorously. These networks are shown to be optimally fault-tolerant. T_n is also shown to possess wide containers with short length. Fault-diameter of T_n is shown to be n. While the T_n can efficiently embed star and bubble-sort graphs, it can also lend itself to an efficient embedding of meshes and hypercubes.

Key Words: Bubble-Sort graph, Cayley Graph, Embedding, Fault Diameter, Fault Tolerance, Generator, Permutation, Star Graph, Transposition.

1 Introduction

In this paper we address the problem of augmenting interconnection network topologies with extra links such that resulting network can tolerate a given number of arbitrary link failures in the sense that the remaining (surviving) network still has the original topology as a subgraph. Several researchers [Lat91, BCH93] have considered adding extra dimensions to networks for tolerating link failures. In [LEA89], an architecture called *folded hypercube* or FHC was analyzed which is basically a hypercube augmented by a single dimension and it was shown that this extra dimension can replace any existing dimension of the network to form a new hypercube of the same size making the network tolerant of any single link failure. In [BCH93], the authors have investigated the idea of adding wildcard dimensions to meshes and tori, and they have constructed dimensionally-augmented meshes and tori.

Cayley graphs in general and star graphs in particular have received much attention since their introduction in 1987 [AK87b]. Vertex and edge symmetry, hierarchy, sublogarithmic diameter and degree of the vertex, and high resilience all contribute to the popularity of the star graph as an attractive alternative to hypercubes for large interconnection networks [DT94, QMA92]. Star graphs accommodate more vertices with less interconnection hardware and less communication delay and many parallel algorithms can be efficiently mapped on star graphs [MS90, FA91, MS92]. Our purpose here is to develop a method to add wildcard dimensions to these star networks. We show that for $n > 4$, $(2n-4)$ extra generators are necessary to have a single wildcard generator in the augmented star graph. This result is generalized to obtain the optimal number of extra generators required to obtain a star graph S_n with some k wildcard generators, for any k, $k \leq \lceil n/2 \rceil$. Interestingly, for the maximum number of wildcard generators, the augmented star graph is found to coincide with the *Transposition Network*, mentioned in [Lei90]. We then show that this new Transposition network can also be efficiently used to provide maximum number of wildcard dimensions to *Bubble-sort* graphs [AK89], another class of interesting Cayley graphs. It appears that the different properties like low diameter, strong resilience (vertex and link fault tolerance) enjoyed by the Cayley graphs are dictated by the intrinsic properties of their respective generator sets. The basic nature of the "transposition" operation plays an important role towards contributing to the richness of the star graphs and other families of graphs that use transpositions as their generators. So, in the second part of the paper, we investigate in detail the algebraic properties of this transposition graph (where the generator set includes all possible transpositions). We show that these transposition graphs are maximally fault tolerant and have wide containers of very short lengths and hence they compare favorably with those reported in [MP88].

2 Background

A Cayley graph is a vertex-symmetric graph often with $n!$ vertices, each vertex denoting a distinct permutation of the set $\{1, 2, \ldots, n\}$. The adjacency among the vertices is defined based on a set of permutations referred to as *generators*; the neighbors of a vertex are obtained by composing the generators with the label of the vertex. The

set of generators defined for a Cayley graph are generally closed under the inverse operation to guarantee that the graph is undirected. Furthermore, the set of generators together with the identity permutation must produce all $n!$ elements of the permutation group Π_n, or a subgroup of Π_n when the generators are repeatedly applied to the already generated vertices. The action of a generator on a permutation label may take various forms. A generator may swap two digits in positions i and j in the label; this action is commonly called a *transposition* and the corresponding generator is designated by $g(i,j)$. The focus of this paper is the graphs whose generators are transpositions; the popular networks with this property are: star graphs and bubble-sort graphs.

Definition 2.1 *A Star graph S_n of dimension n is a Cayley graph with $n!$ vertices, each labeled with a distinct permutation of the set of integers $\{1,\ldots,n\}$. The set G of generators in S_n is defined as: $G = \{g(i,j), j = 1,2,\ldots,(i-1),(i+1),\ldots,n\}$. Without loss of generality, it is assumed $i = 1$.*

The topological properties of the star graph have been derived and discussed elsewhere [AK87b, AK87a, DT91, Kav93, Lat93, SS92, SS91]. Briefly, S_n is both vertex and edge transitive. It contains $n!(n-1)/2$ links with a diameter of $\lceil 3(n-1)/2 \rceil$.

Definition 2.2 *A Bubble sort graph B_n of dimension n is a Cayley graph with $n!$ vertices, each labeled with a distinct permutation of the set of integers $\{1,\ldots,n\}$. The set G of generators in B_n is defined as: $G = \{g(i,i+1), i = 1,2,\ldots(n-1)\}$.*

The topological properties of bubble sort graphs can be found in [AK89, Knu73]. Briefly, B_n is vertex and edge transitive, is regular with vertex degree $(n-1)$, and has $n!(n-1)/2$ links with a diameter of $n(n-1)/2$.

3 Wildcard Dimensions in S_n and B_n

Both star graphs and bubble-sort graphs belong to the class of hierarchical networks whose sets of links can be partitioned to dimensions. Examples of other such networks include meshes, tori and hypercubes. For such networks, any wildcard dimension can be utilized to replace any of the original dimensions such that the faulty graph (with link failures) still contain the original graph; this is different from the fault tolerance of the original graphs. In [BCH93], the authors have investigated the idea of adding wildcard dimensions to -dimensional meshes and tori. Our purpose is to add wildcard to star graphs and bubble-sort graphs.

3.1 Augmentation of the Star graph

In order to find a solution to an optimal augmentation of dimensions in star graph, we first show that the generators of this graph are unique upto isomorphism among digits used.

generator	*cycle representation*	*generator*	*cycle representation*
1234	*null*	3124	(123)
1243	(34)	3142	(1243)
1324	(23)	3214	(13)
1342	(243)	3241	(143)
1423	(234)	3412	(13)(24)
1432	(24)	3421	(1423)
2134	(12)	4123	(1234)
2143	(12)(34)	4132	(124)
2314	(132)	4213	(134)
2341	(1432)	4231	(14)
2413	(1342)	4312	(1324)
2431	(142)	4321	(14)(23)

Table 1: Possible Generators for S_4

Lemma 3.1 *Let $g(i,j)$ and $g(i,k)$ be two generators of a Cayley graph S' (with $N = n!$ vertices), where $1 \leq i \neq j \neq k \leq n$ and $n > 3$. For S' to be isomorphic to S_n it is necessary and sufficient that all the remaining generators of S' be of the form $g(i,m)$ where $1 \leq m \leq n$, and $m \neq i,j,k$.*

Proof : The proof is by induction. The claim is certainly true for S_4. Suppose it is true for S_{n-1}. Since S' is isomorphic to S_n, it must include n S_{n-1}'s as its subsets. Thus at least $(n-2)$ of the generators of S' must be of the form $g(i,x)$. In that case the last generator must also be of the form $g(i,x)$, otherwise S' cannot be isomorphic to S_n. □

Lemma 3.1 provides a basis for proving that the generators of any graph isomorphic to S_n must necessarily be of form $g(i,x)$. To see this, the special case of S_4 is studied first. All the possible permutations that can be defined as generator candidates for a graph S' isomorphic to S_4 are shown in Table 1.

The permutations in Table 1 can be classified into 4 conjugacy classes depending on the structure of their cycle representations.

Class I (a single 2-cycle): $(12),(13),(14),(23),(24),(34)$

Class II (a single 3-cycle): $(123),(132),(234),(243),(124),(134),(142),(143)$

Class III (a single 4-cycle): $(1234),(1243),(1324),(1342),(1423),(1432)$

Class IV (two 2-cycles): $(12)(34),(13)(24),(14)(23)$

From edge-transitivity of the star graph, it follows immediately that the set of generators of S' must belong to the same class. Consider Class I. It is impossible to pick 3 generators in this class for S' such that the condition specified

in Lemma 3.1 is not met. The generators cannot be picked from Classes II and III simply because for any subset (of order 3) of these classes the closure condition with respect to inverse operation cannot be held. For example in Class II, if (123) is selected as a generator, then $(123)^{-1} = (132)$ must also be included.[1] But to complete this set, there is no other distinct generator in Class II which is self-inverse. In Class IV, it can be easily shown that the existing 3 generators cannot produce a graph isomorphic to S_4 (for instance, try to from a ring of length 6 using an alternate sequence of two generators). We summarize this in the next theorem.

Theorem 3.1 *Let S' be a graph isomorphic to S_n. Then its set of generators must be of the form $g(i, x)$ where $1 \leq i \leq n$, and $x \in \{1, 2, \ldots, n\} - \{i\}$.*

Suppose it is desirable to construct a network based on S_n which can tolerate any i link failures with no degradation in performance. This is equivalent to supporting the network with i wildcard dimensions. One direct approach is to increase the number of links in every dimension by a multiplicative factor of $(i + 1)$ (replication). Consider the S_n with the set of generators: $g(1,2)$, $g(1,3)$ $\ldots$ $g(1,n)$. To produce one ($i = 1$) wildcard generator, one has to add only transpositions or 2-cycles to the above set (by Theorem 3.1). The optimal set of generators required is:

$$\begin{array}{cccccc} - & g(1,2) & g(1,3) & \ldots & g(1,n) \\ g(2,1) & - & g(2,3) & \ldots & g(2,n) \\ g(3,1) & g(3,2) & - & \ldots & g(3,n) \end{array}$$

Any row in the above corresponds to the set of generators which can produce S_n and any link failure can damage at most two of the above 3 rows (when $g(1,2)$ or $g(1,3)$ fails). Thus any single link-failure can be tolerated. The number of generators required can be easily obtained as : $(3n - 6)$. The direct approach (doubling each dimension or replication) requires a total of $(2n - 2)$ generators which is *better* than the above method for $n > 4$. This result is summarized in the following.

Remark 3.1 *Unlike the hypercube, the star graph cannot achieve the single-wildcard feature by adding only one dimension.*

[1] Another argument to rule out Class II is that its generators cannot change the parity of the starting vertex.

$dimension$	$no.\ of\ wildcards$ i	$Replication\ Method$	$Proposed\ Method$
$n = 2k$	$(k-1)$	$k(2k-1)$	$k(2k-1)$
$n = 2k+1$	k	$2k(k+1)$	$k(2k+1)$

Table 2: Comparison of Two Methods

For $i > 1$, one needs to add two rows to the existing rows for each extra wildcard since the failure of one generator can affect 2 rows. As the number of rows increases, the number of required generators for new rows drops since some of these generators have already been included in the generator set. The total number of generators needed to obtain i wildcards $1 \leq i \leq (\lceil \frac{n}{2} \rceil - 1)$ is $(i+1)(n-1)$ using $Replication\ Method$ and is $(2i+1)(n-i+1)$ using the $Proposed\ Method$.[2] It follows that for $i > \lceil \frac{n-2}{2} \rceil$ the proposed method proves slightly advantageous; see Table 2. Note that to provide maximum number of wildcards the number of generators for even n is the same for both cases and for odd n saving in generators using the proposed method is k. However, routing doesn't change with replication.

Now, if we consider the generators of the augmented star with $(\lceil n/2 \rceil - 1)$ wildcard dimensions, the augmented graph obtained as such provides all the possible transpositions on n available dimensions (i.e. defined by $\frac{n(n-1)}{2}$ generators). This graph was introduced in [Lei90] and briefly mentioned in [LJD93] as the *Transposition graph*, T_n.

Definition 3.1 *A Transposition graph T_n, of dimension n, is a Cayley graph with $n!$ vertices each assigned a distinct permutation of the first n positive integers. The set of generators is: $\{g(i,j), 1 \leq i,j \leq n, i \neq j\}$.*

Example: Figure 1 shows transposition graphs of dimensions 2 and 3 and Figure 2 shows a transposition graph of dimension 4. Note that T_n is a regular vertex symmetric undirected graph with vertex degree $\frac{n(n-1)}{2}$, has $n!$ vertices and $n(n-1)n!/4$ edges.

[2] The maximum value for i is dictated by the fact that all the generators will be used up for i_{max} and any further improvement must come from replication of already used generators.

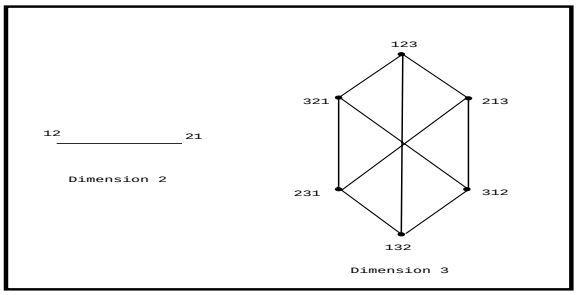


Figure 1: Transposition Graphs of Dimensions 2 and 3

3.2 Augmentation of the Bubble-Sort graph

In this section we investigate the possibility of adding wildcard generators to the bubble sort graph B_n and show that for addition of maximum number of wildcard generators the augmented graph is again the transportation graph T_n, as obtained for star graphs.

Lemma 3.2 *It is possible to add one generator to the set of generators of B_n which can play the role of a wildcard generator.*

Proof : The proof is by construction. Let the set of generators in B_n be: $\{g(i_1, i_2), g(i_2, i_3), \cdots, g(i_{n-1}, i_n)\}$. The added generator will be: $g(i_n, i_1)$. So the augmented set of generators will be: $\{g(i_1, i_2), g(i_2, i_3), \cdots, g(i_{n-1}, i_n), g(i_n, i_1)\}$. It can be seen that through the newly added wrap around generator, failure of any single generator can be tolerated (note that B_n is edge-transitive [AK89]). □

Since the Bubble-sort graph is edge-transitive, the generators can be arranged arbitrarily with respect to digits. The transportation graph T_n contains all possible transposition generators and hence T_n contains many B_n's. How many distinct B_n's are contained in a T_n? One can easily establish a one-to-one correspondence between the set of generators of a B_n with a specific permutation of n digits. Let $\sigma = (i_1, i_2 \ldots i_n)$ correspond to the set of generators (for a B_n): $\{g(i_1, i_2), g(i_2, i_3) \ldots g(i_{n-1}, i_n)\}$. We have $n!$ distinct permutations of n digits; but any permutation corresponds to the same set of generators as its reflection would. For instance permutations: $(i_1, i_2 \cdots i_n)$ and $(i_n \cdots i_2, i_1)$ correspond to the same set of generators. We state the following theorem.

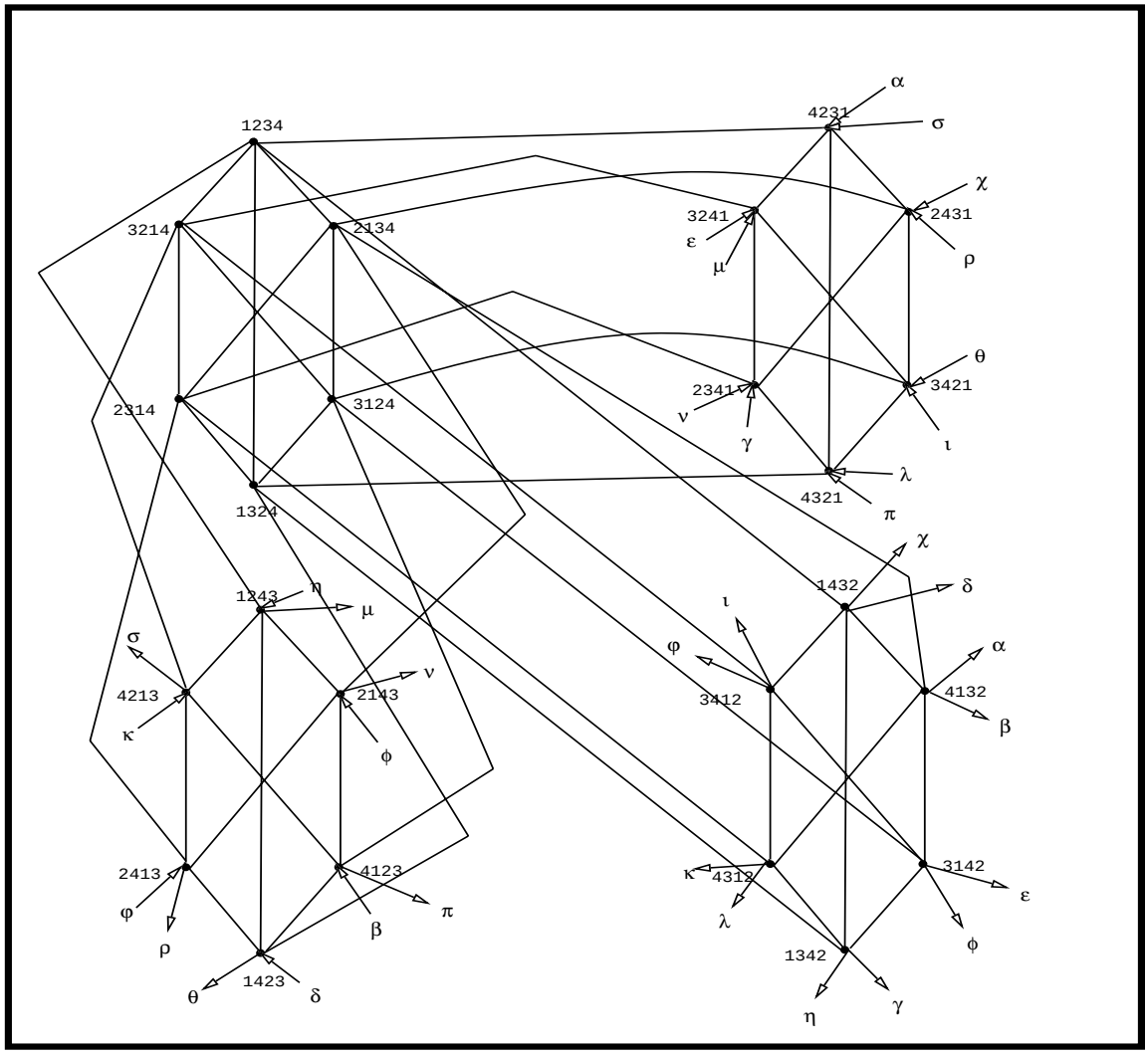


Figure 2: Transposition Graph of Dimension 4

Theorem 3.2 *The set of generators of T_n contains $n!/2$ generator subsets of cardinality $(n-1)$, each subset specifying uniquely a B_n.*

Remark 3.2 *These B_n's are distinct, but not necessarily mutually edge-disjoint.*

The next important question to be answered is: how many wildcard generators are contained in a T_n if it is to be used as a fault tolerant Bubble-sort graph, i.e., how many link failures can a T_n tolerate and can still contain a B_n? We introduce the simple concept of a *model graph* H_n corresponding to any graph (say, T_n or B_n) whose generators are transpositions in the following way: H_n has n vertices, each labeled with the digits 1 through n and two vertices

i and j are connected by an undirected edge if and only if the generator $g(i,j)$ is included in the generator set of the graph H_n is modeling. Thus, the model graph of T_n is a complete graph of n vertices, and that of B_n is a line graph of n vertices. When any link fails in a T_n, we can model the situation by deleting an edge (i,j) in the model graph (the generator $g(i,j)$ corresponding the failed link in T_n). Thus, as long as we can find a Hamiltonian path in the remaining model graph of T_n (corresponding to any fault scenario), a complete bubble sort graph can be constructed from the remaining fault free links of the transposition graph T_n. In case of arbitrary failure of links in T_n, the model graph is an arbitrary graph of n vertices and the problem reduces to finding a Hamiltonian path in the model graph. Computing a Hamiltonian path in an arbitrary graph is NP-complete in general [GJ79]; so is the problem of locating a B_n (if any) from an arbitrarily injured T_n. Nevertheless for special cases when the number of failures is restricted, one may be able to obtain a valid set of generators in linear time.

Theorem 3.3 *T_n has at most $(n-2)$ wildcard generators as a fault tolerant B_n.*

Proof : If generators of any row in the transposition matrix fail (there are $(n-1)$ of such generators), no B_n can be constructed by the remaining generators. This is so since in the model graph H, all links incident to a vertex would be missing, isolating this vertex from the rest of H. □

Theorem 3.4 *Failure of any arbitrary set of at most $(n-2)$ generators cannot destroy all the eligible candidates for the set of generators of B_n.*

The number of disjoint generator sets of T_n which can be used for construction of a B_n is equivalent to the number of link-disjoint Hamiltonian paths in K_n (a fully-connected n-vertex graph). This number can be at most $\lfloor n/2 \rfloor$.

4 Properties of the Transposition Graph T_n

The transposition graph T_n, for any $n \geq 1$, is a Cayley graph and hence is vertex symmetric [AK89]. So, in T_n, we can always view the distance between any two arbitrary vertices as the distance between the source node and the identity permutation by suitably renaming the symbols representing the permutations. Hence, in our subsequent

discussion about a path from a source vertex to a destination node, the destination vertex is always assumed to be the identity vertex I without any loss of generality. Also, we use distance of a vertex (permutation) to indicate its distance to the identity vertex.

It is easy to see that any permutation of n elements can also be specified in terms of its cycle structure with respect to the identity permutation I. For example, $345216 = (135)(24)(6)$. The maximum number of cycles in a permutation of n elements is n and the minimum number is 1. When a cycle has only one digit, that digit is in its correct position in the permutation with respect to the identity permutation: we call such a digit an *invariant* digit. The singleton cycles may be omitted in the cycle representation of a permutation if the number of digits in the permutation is understood from the context. We use following notations throughout the rest of the paper: Δ_n, Diameter of T_n, u, An arbitrary permutation (vertex), $(C_1 \cdots C_k)$, Cycle representation of the arbitrary vertex u; C_i^j denotes j-th digit in the i-th cycle of vertex u, $D(u)$, Distance of the vertex u (from the identity permutation), $\mu(u)$, Number of cycles of length at least 2 in any permutation u, $\theta(u)$, Number of cycles of length 2 in any permutation u, $m(u)$, Total number of digits in these μ cycles of the permutation u, and $n - m(u)$, Number of invariant digits in the permutation u.

4.1 Shortest Routing and Diameter

We want to reach the identity vertex I starting from an arbitrary vertex u in T_n. Consider the algorithm: "For each digit in u not in correct position, put it in correct position by the appropriate generator."

Remarks: (a) The algorithm is well-defined; for each vertex in T_n we have all possible $n(n-1)/2$ transpositions (generators). (b) Consider an arbitrary cycle C_s, $|C_s| \geq 2$, in the vertex (permutation) u; we need exactly $|C_s| - 1$ moves of the algorithm to execute the cycle completely (once $|C_s| - 1$ digits of the cycle C_s are placed in correct positions, the last one is automatically placed in its position). (c) For an arbitrary vertex u in T_n, the algorithm makes $m(u) - \mu(u)$ moves to reach the destination vertex I. (d) For an arbitrary vertex u in T_n, at least $m(u) - \mu(u)$ moves are necessary to reach the vertex I since at most one out-of-place digit (belonging to a cycle of length > 2) can be placed in its correct position in one move (for 2-cycles, when one out-of-place digit is put in its correct position, the other is also automatically put in its correct position).

Theorem 4.1 *The distance of an arbitrary vertex u in T_n (from the identity vertex I) is given by $D(u) = m(u) - \mu(u)$ and the diameter of T_n is given by $\Delta_n = n - 1$.*

Corollary 4.1 *Each vertex in T_n has $(n-1)!$* **antipodes**, *i.e., vertices at maximum distance from the vertex. For example, for the identity vertex I, these vertices are those having the cycle representation of the form $(1*)$ where $*$ is any arbitrary permutation on the integers of the set $\{2, 3 \ldots, n\}$.*

4.2 Average Distance of T_n

The average number of cycles in a permutation of n digits is known to be $H_n = \sum_{i=1}^{n} 1/i$, the nth harmonic number. Thus the total number of cycles over all permutations is: $n!H_n$. On the other hand, the total number of m-cycles is: $n!/m$, $1 \leq m \leq n$. But the total number of cycles times the average length of a cycle ℓ_{av} should equal to the sum of all m-cycles times their number. Therefore,

$$n!H_n \times \ell_{av} = \sum_{m=1}^{n} \frac{n!}{m} m \quad \Rightarrow \ell_{av} = \frac{n}{H_n}$$

The average number of cycles in a permutation is: H_n. Hence:

$$\text{Average Distance in } T_n = \left(\frac{n}{H_n} - 1\right) \times H_n = n - H_n$$

4.3 Fault Tolerance of T_n

Lemma 4.1 *For an arbitrary vertex u in T_n there exist at least $(m(u) - \theta(u))$ vertex disjoint paths (to the identity vertex I) of length $D(u) = m(u) - \mu(u)$.*

Proof : The minimal length is easily established by noting that any cycle of length k can be executed using $(k-1)$ transpositions. Let $A = \{i_1 i_2 \ldots i_{(m-2\theta)}\}$ be the set of out-of-place digits belonging to k-cycles, $k > 2$. In addition, let $B = \{j_1 j_2 \ldots j_{(2\theta-1)} j_{(2\theta)}\}$ be the set of out-of-place digits belonging to 2-cycles. Disjoint paths can be obtained by changing the order in which digits are corrected. Pick the initial order of correction as: $\sigma = (i_1 i_2 \ldots i_{(m-2\theta)} j_1 j_2 \ldots j_{(2\theta-1)} j_{(2\theta)})$. By cyclically shifting σ to the left every time, a distinct disjoint path will be obtained. The correction order corresponding to these paths are as follows: $\sigma_1 = (i_1 i_2 \ldots i_{(m-2\theta)} j_1 j_2 \ldots j_{(2\theta-1)} j_{2\theta})$,

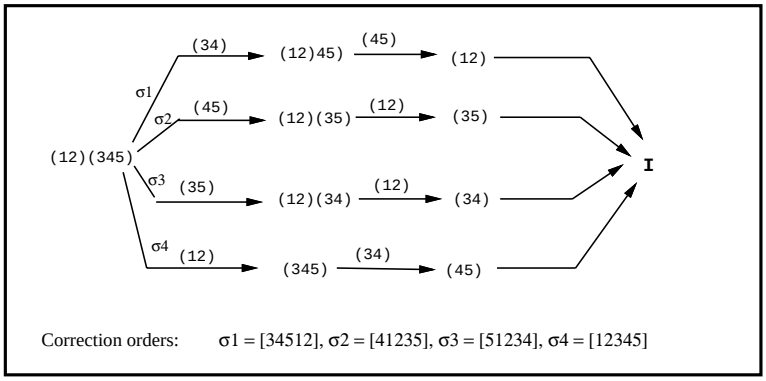


Figure 3: Vertex Disjoint Minimal Paths in T_6

$\sigma_2 = (i_2 i_3 \ldots i_{(m-2\theta)} j_1 j_2 \ldots j_{2\theta} i_1), \cdots, \sigma_{(m-2\theta+1)} = (j_1 j_2 \ldots j_{2\theta}, i_1 i_2 \ldots i_{(m-2\theta)}), \sigma_{(m-2\theta+2)} = (j_3 j_4 \ldots j_{(2\theta)}, i_1 i_2 \ldots i_{(m-2\theta)} j_1 j_2), \cdots, \sigma_{(m-\theta)} = (j_{(2\theta-1)} j_{(2\theta)} \ldots j_{(2\theta-3)} j_{(2\theta-2)})$.

Observe that starting from $\sigma_{(m-2\theta+1)}$, the new correction orders are obtained by shifting two positions at a time (because every pair of adjacent elements in B corresponds to a 2-cycle and correction of one element results in the correction of the second one). □

Example: Consider the vertex $u = (12)(345)(6)$ in T_6. Here, $m = 5, \mu = 2, \theta = 1$. Thus, there are 4 vertex disjoint paths, each of length 3, from vertex u to the identity I. These paths, along with the correction order of digits for each path, are shown in Figure 3.

For an arbitrary vertex u in T_n, we can apply any of the possible $n(n-1)/2$ generators (transpositions); only $m(u) - \theta(u)$ of them lead to vertex disjoint minimal paths. In order to determine the vertex connectivity of T_n, we need to explore also the non minimal vertex disjoint paths. First, we observe certain simple facts:

Remark 4.1 *Any k-cycle (cycle of length k) in the cycle representation of any vertex in T_n can be written in k different ways. For example, the cycle (2546) is the same as the cycles (5462), or (4625), or (6254).*

Remark 4.2 *Consider a k-cycle in any vertex u, say $(i_1 i_2 \cdots i_k)$. There exist $\binom{k}{2}$ generators to break this cycle. Application of the generator $g(i_j, i_{j'})$, $j < j'$ (for brevity), breaks the cycle into two cycles (each of length at least 2): $(i_1 i_2 \cdots i_{j-1} i_{j'} i_{j'+1} \cdots i_k)\ (i_j i_{j+1} \cdots i_{j'-1})$.*

Remark 4.3 *Consider an arbitrary vertex u in T_n, $u = (C_1)(C_2)(...) = (C_1^1 \cdots C_1^k)(C_2^1 \cdots C_2^{k'})(...)$. By applying the generator $g(C_1^1, C_2^1)$ we reach the vertex $(C_1^1 \cdots C_1^k C_2^1 \cdots C_2^{k'})(...) = (C_2^2 \cdots C_2^{k'} C_1^1 \cdots C_1^k C_2^1)(...)$ (see Remark 4.1). Note that (...) indicates other cycles in the vertex, if any.*

Lemma 4.2 *Consider an arbitrary vertex u in T_n, $u = (C_1)(C_2)(...) = (C_1^1 \cdots C_1^k)(C_2^1 \cdots C_2^{k'})(...)$. There are $k \times k'$ distinct neighbors of the vertex u each of the form $(*xy)(...)$ where x and y are arbitrary digits such that $x \in C_1$ and $y \in C_2$ and $*$ indicates some digit string.*

Proof : The proof follows from the Remark 4.3 and the fact that $|C_1| = k$ and $|C_2| = k'$. Note that for any choice of x (say, $x = C_1^i$) and y, the generator needed to reach the neighbor is given by $g(C_1^{i+1}, y)$. Distinctness follows from the distinctness of the generators for different choices of x and y. □

Lemma 4.3 *Consider an arbitrary vertex u in T_n, $u = (...)(C_s^1 C_s^2 \cdots C_s^k)$, such that $k > 3$ (C_s is an arbitrary cycle of k digits). There exist $\binom{k}{2} - k$ vertices each of the form $(...)(*C_s^i C_s^j)$, $1 \leq i \leq k-2$ and $i+2 \leq j \leq k$ (the combination $i = 1, j = k$ excluded), each at a distance 2 from the vertex u. Moreover, these vertices as well as the vertices on the paths from u to these vertices are all distinct.*

Proof : Each pair of digits C^i, C^j is distinct and the two required generators (transpositions) needed for each choice of C^i, C^j are $g(C^i, C^j)$ and $g(C^{i+1}, C^j)$ (also see Remark 4.1). Note that none of these vertices involved are on any minimal path from u to the identity vertex I. □

Remark 4.4 *Let $(C_1^1 C_1^2 \ldots C_1^k)$ be the cycle representation of an arbitrary vertex with a single cycle C_1. Applying the sequence of transpositions $(C_1^1 C_1^2)$, $(C_1^2 C_1^3)$, $\ldots (C_1^{k-1} C_1^k)$ to this vertex and in that order will map the vertex to identity. Furthermore, at step j, digit C_1^j is put in correct position as the result of transposition (C_1^j, C_1^{j+1}), $1 \leq j \leq k-1$. For a given cycle, this routing is referred to as the* **standard minimal routing** *for that cycle.*

Lemma 4.4 *Let u and v be a pair of vertices in T_n such that u has a cycle of the form $(\alpha x y)$ (possibly other cycles as well) and v has a cycle of the form $(\alpha w z)$ (possibly other cycles as well), where x, y, z, and w are single digits and α is a string of 1 or more digits. Let P be the set of shortest paths from u to the identity vertex I such that the generator $g(x, y)$ is applied last (standard minimal routing). Similarly, let Q be the set of shortest paths from v to*

the identity vertex I such that the generator $g(w, z)$ is applied last (standard minimal routing). Any path in P is vertex disjoint from any path in Q iff $\{x, y\} \neq \{w, z\}$.

Proof : The digit y occupies the correct position of the digit x for each vertex along every path in P and each path in P passes through the vertex (xy) before reaching the node I. Similarly, the digits z occupies the correct position of the digit w for each vertex along every path in Q and each path in Q passes through the vertex (wz) before reaching the node I. The claim readily follows. □

Theorem 4.2 *For any given vertex u in T_n, there exist $n(n-1)/2$ vertex-disjoint paths (to the identity vertex I) whose lengths are as follows:*

No. of Paths	*Length*
$(m - \theta)$	D
$\binom{n-m}{2} + (n-m) \times m + \binom{m}{2} - (m - \theta)$	$D + 2$ *(at most)*

where $m = m(u)$, $\theta = \theta(u)$, and $D = D(u)$.

Proof : Let the cycle representation of the vertex u be $u = C_1, C_2, \cdots, C_\mu$, where $\mu = \mu(u)$ and $\sum_{i=1}^{\mu} |C_i| = m$.

Denote the set of vertex disjoint minimal paths by Ψ_1. According to Lemma 4.1 $|\Psi_1| = (m - \theta)$. Note that none of the invariant digits in vertex u are ever moved in any of these minimal paths.

There are $(n-m)$ invariant symbols in vertex u. Let $(*)$ denote the cycle representation of the vertex u. Choose any two arbitrary invariant digits, say digits i and j, apply the generator $g(i, j)$ to the vertex u to reach the node $(*)(ij)$, then apply shortest routing to the $(*)$ portion of the vertex and execute the cycle (ij) last. Call the set of paths formed this way Ψ_2, where $|\Psi_2| = \binom{n-m}{2}$. Each path in Ψ_2 is vertex disjoint from any other in Ψ_2 because of the unique choice of pairs of invariant digits; also each path in Ψ_2 is vertex disjoint from any path in in Ψ_1. Each path in Ψ_2 has a length of $D + 2$.

Choose an arbitrary invariant digit α (there are $(n - m)$ choices) and an arbitrary digit $\beta = C_i^j$ from any cycle C_i (there are μ choices for i and $|C_i|$ choices for j); apply the generator $g(\alpha, \beta)$ to the vertex u to reach the node $C_1, C_2, \cdots, C_{i-1}, (C_i^{j+1} \cdots C_i^{|C_i|} C_i^1 \cdots \alpha\beta), C_{i+1} \cdots C_\mu$ (Remark 4.3), apply shortest routing to reach the identity vertex using the generator $g(\alpha, \beta)$ last. Let Ψ_3 denote the set of paths generated according to this scheme. Each

path in Ψ_3 has a length of $D+2$ and $|\Psi_3| = \sum_{i=1}^{\mu}(n-m)|C_i| = m(n-m)$. Paths in Ψ_3 are mutually disjoint by the unique choices of α and β for each path; they are also vertex disjoint from the paths in Ψ_1 and Ψ_2.

There are θ cycles of length 2 and $\mu - \theta$ cycles of length > 2 in vertex u. We generate two sets of non minimal paths. First, for each k-cycle, $k > 2$, we go to the $\binom{k}{2} - k$ distinct neighbors of u (each at a distance 2 from u) as defined in Lemma 4.3 and from each of these vertices follow *standard minimal routing* to reach the identity vertex I. Each path is of length at most $D+2$ and they are mutually vertex disjoint (Lemma 4.4). Second, choose any pair of cycles in u, generate distinct neighbors as in Lemma 4.2 and follow *standard minimal routing* from each neighbor to reach the identity vertex I. Each path is of length $D+2$ and they are mutually vertex disjoint (Lemma 4.4). Let Ψ_4 denote the set of paths generated according to these two schemes. Each path in Ψ_4 has a length of at most $D+2$ and $|\Psi_4| = \binom{m}{2} - (m-2\theta) - \theta = \binom{m}{2} - m + \theta$ (number of transpositions available involving all m digits in all the cycles is $\binom{m}{2}$; we need to exclude k transpositions for each k-cycle, $k > 2$, (total $(m-2\theta)$ transpositions) corresponding to adjacent digits (leading to minimal paths) and θ optimal transpositions corresponding to 2-cycles). Each path in Ψ_4 is disjoint to any path in Ψ_1 by Lemma 4.4 and is also disjoint from any path in Ψ_2 or Ψ_3 since no invariant is involved.

Note that $|\Psi_1| + |\Psi_2| + |\Psi_3| + |\Psi_4| = \binom{n}{2}$. □

Corollary 4.2 *Vertex connectivity of T_n is $n(n-1)/2$.*

Example: Consider the vertex $(12)(345)(6)$ in T_6. The minimal paths (i.e., paths in Ψ_1) were shown in Figure 3. The set Ψ_2 is empty since there is only one invariant digit e.g. "6". The set Ψ_3 has five paths that are generated by mixing the invariant digit with the 2-cycle and the 3-cycle; these paths are shown in Figure 4. There is no cycle of length > 3; hence the paths in the set Ψ_4 are generated by combining the 2-cycle and the 3-cycle in all possible (six) ways; the paths are shown in Figure 5.

Theorem 4.3 *The fault-diameter of T_n is at most $(n+1)$.*

Proof : According to Theorem 4.2 there are $\binom{n}{2}$ disjoint paths between any two vertices in T_n of length at most $D+2$. In addition, the connectivity of T_n is $\binom{n}{2}$. Therefore, removal of $\binom{n}{2} - 1$ vertices from T_n will leave at least

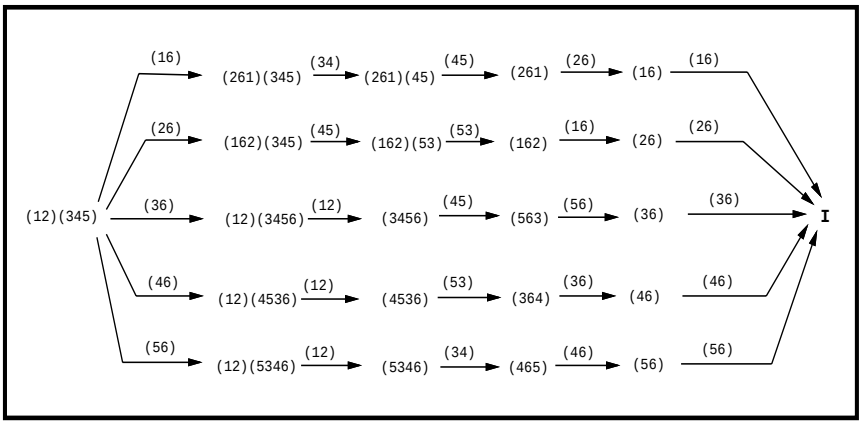


Figure 4: The set Ψ_3 of paths for the vertex $(12)(345)(6)$ in T_6

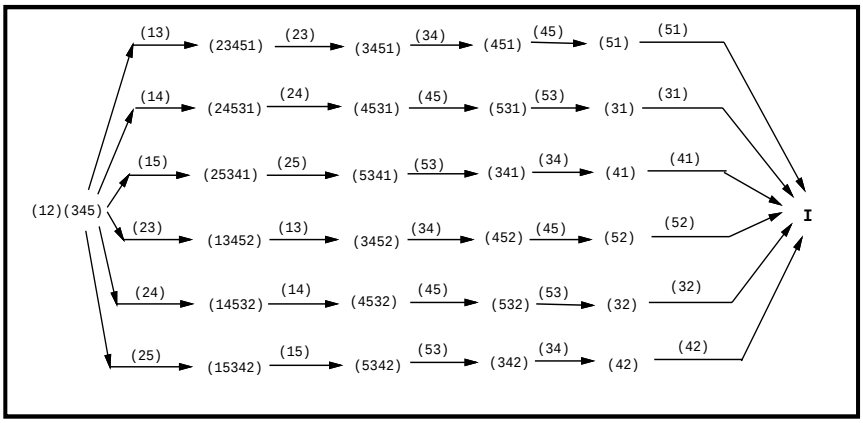


Figure 5: The set Ψ_4 of paths for the vertex $(12)(345)(6)$ in T_6

one disjoint path between two vertices which is longer than the minimal path by an additive 2. The diameter of T_n is $(n-1)$. So the fault-diameter $d_f \leq (n-1)+2 = n+1$.

Let u and v be two vertices in T_n with respective cycle representations of $(1)(23\ldots n)$ and $(12\ldots n)$. Now distribute the $\binom{n}{2}-1$ faults such that all neighbors of u are faulty except for v. Any path from u to identity must pass through v which is an antipode of identity. The length of such a path is $(n-1)+1 = n$. It follows: $n \leq d_f \leq n+1$.

□

Theorem 4.4 *The fault diameter of T_n is n.*

Proof : For brevity, we provide only a sketch of the proof which is established by construction. We have to concentrate on the vertices which are at a maximum distance from the identity vertex (i.e., at a distance of $n-1$). Note that such vertices must have a single cycle containing all the n digits. For such vertices we give a formulation of $\binom{n}{2}$ vertex-disjoint minimal paths (of length $n-1$) to the identity vertex.

Without loss of generality, let the cyclic representation of an antipode u with respect to I be: $(123\ldots n)$. We create an ordered set of all possible transpositions as follows: $t_{SET} = \{(12),(13)\ldots(1n),(23),(24),\ldots(2n),\ldots (1n),(2n),\ldots(n-1,n)\}$. Each optimal disjoint path originates from u via a distinct transposition and terminates at I through another distinct transposition. For any path, if $t_i \in t_{SET}$ is chosen to be the last transposition from u to I, $t_{i+1} \in t_{SET}$ will be selected as the first transposition used along that path. For $\binom{n}{2}$ paths constructed this way, all the neighbors of u and I will be distinct (because there are $\binom{n}{2}$ t_i's in t_{SET}). Depending on t_i, two cases are distinguished:

Case(i): $t_i = (jn), 1 \leq j \leq n-2$. Clearly applying $t_{i+1} = (j+1, j+2)$ automatically corrects digit $j+1$. If $t_i = (n-1,n)$, then $t_{i+1} = (12)$ which after being applied to u will correct digit 1. Thus, starting from digit $j+1$ (or 1), other digits (i.e. $j+2, j+3, \ldots$) are corrected in a cyclic order leaving out digits j and n which will get corrected in the last step. *Case(ii)*: $t_i = (jk), j \neq k \neq n$. Here $t_{i+1} = (j, k+1)$. We begin by correcting digit $k+1$ and the digit correction order will be $(k+1, k+2, \ldots j-1, j+1, \ldots, k-2, k-1)$. Digits j and k will get corrected in the last step.

We show the distinctness of the intermediate vertices along the paths constructed as described before by contradiction. Suppose two paths between u and I meet at an intermediate vertex q before reaching I. For this to happen, the same set of digits would have to be corrected in the same number of steps in the above paths, no matter what the correction order is. But this cannot happen due to the cyclic nature of the correction order used in construction of each path and the uniqueness of the pair of digits which will get corrected in the last step along each path. □

Example: Consider the vertex (123456) in T_6. $t_{SET} = \{(12),(13),(14),(15),(16),(23),(24),$ $(25),26),(34),(35),(36),(45),(46),(56)\}$. All $\binom{6}{2} = 15$ minimal paths, each of length 5, are shown in Figure 6.

Remark 4.5 *Another very important consideration in designing robust interconnection networks is the existence of* wide containers *of short lengths [MP88]. A* container *is a set of vertex disjoint paths between arbitrary pair of*

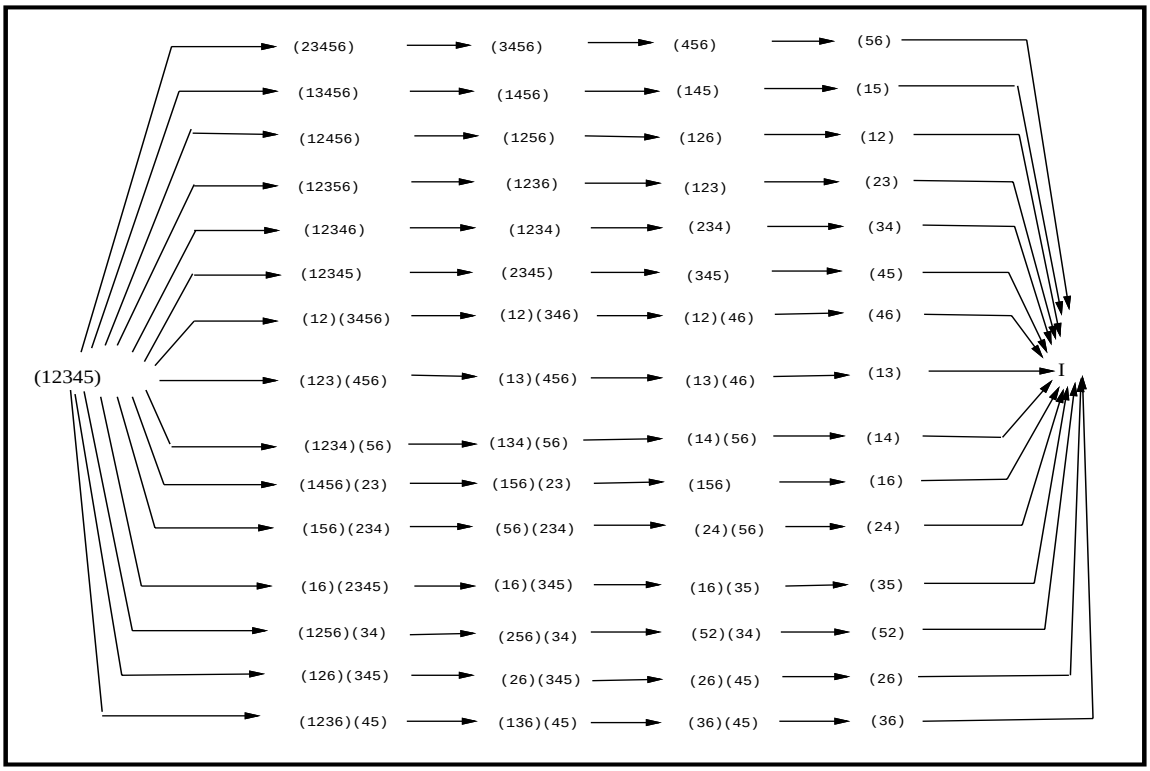


Figure 6: Minimal Vertex Disjoint Paths from the vertex (123456) in T_6

vertices. The existence of wide containers (a large number of vertex disjoint paths) is essential to establish virtual links between vertices both when there are permanent regular faults (vertices just cease to exist) as well as when the faults are Byzantine in nature (i.e., faulty vertices can behave maliciously by changing or misdirecting messages). The details, protocols and algorithms to establish virtual links with the help of containers can be found in [MP88]. From the above theorems, it is apparent that all containers in T_n are of length $D+2 \leq n+1$ and that each container in T_n is of length at most diameter of the graph plus one. Thus these graphs compare very favorably with the set of graphs with wide containers of short lengths presented in [MP88].

4.4 Embedding of Other Graphs in T_n

It is clear that the transposition graph T_n contains star graph S_n and bubble-sort graph B_n i.e. these graphs can be optimally embedded in T_n (with dilation one and expansion one). In this section we explore embeddability of some other networks like hypercubes and 2D meshes in T_n.

Theorem 4.5 *T_n is bipartite; T_n does not contain any cycle of odd length.*

Proof : The proof follows from the fact that the transpositions are odd permutations. Any Cayley graph based on transpositions is bipartite. □

Theorem 4.6 *T_n ($n > 2$) contains all cycles of even length ℓ, $4 \leq \ell \leq n!$.*

Proof : It has been shown in [JLD91] that a star graph S_n contains all ℓ cycles for even ℓ and $6 \leq \ell \leq n!$. Since S_n is a subgraph of T_n, the result readily follows. □

Theorem 4.7 *A Transposition graph T_n contains $\binom{n-1}{k}\binom{n}{k}k!n$ distinct S_k's (S_k is a star graph of dimension k).*

Theorem 4.8 *In T_n, there are $\lfloor \frac{n}{2} \rfloor$ disjoint sets of generators for B_n.*

Theorem 4.9 *Consider a $M_1 \times M_2$ 2-dimensional mesh such that $M_1 \times M_2 = n!$. There is an optimal (expansion = 1) embedding of this mesh in T_n.*

Proof : It has been shown in [JLD91] that a $M_1 \times M_2$ 2-dimensional mesh such that $M_1 \times M_2 = n!$ can be embedded in a star graph S_n with unit expansion and dilation 3 where edges with dilation 3 correspond to transpositions of the form $(i, j), i, j \neq 1$ which can be executed using three generators in S_n namely $(1i), (1, j), (1, i)$. But in T_n transpositions of the form (i, j) can be executed with only one generator (since all possible transposition generators are present). Hence the result follows. □

Example: Figure 7 shows the mapping of a 6×4 mesh in T_4. Note that all the vertices are not labeled and the edge label (ij) indicates that the generator $g(i, j)$ is used.

Next, we consider embedding of hypercubes in transposition networks. It is to be noted that the embedding of hypercubes into star graph has been investigated before (see [NSK90] for instance). These embeddings generally offer dilations of 3 or more. For Transposition networks, on the other hand, more efficient embedding is expected to exist due to their robust structure.

Theorem 4.10 *There is a dilation-1 embedding of hypercube Q_{n-1} into T_n.*

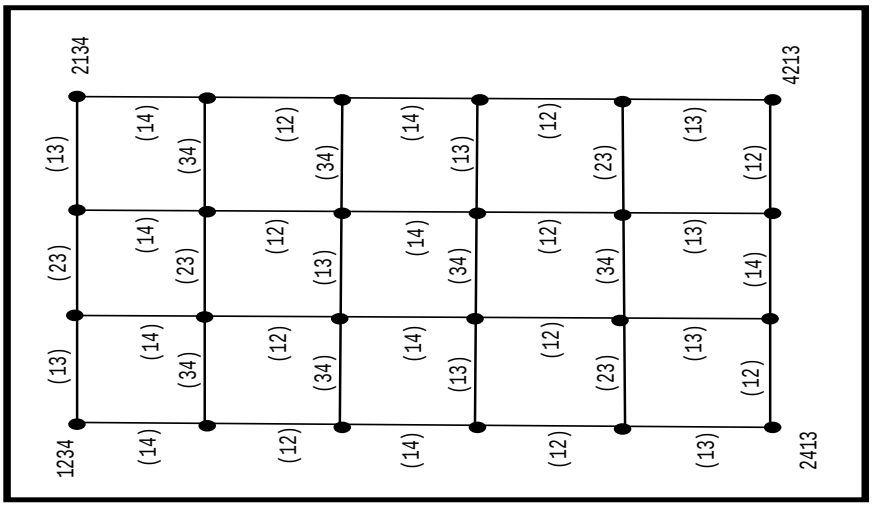


Figure 7: Mapping of a 6×4 mesh in T_4

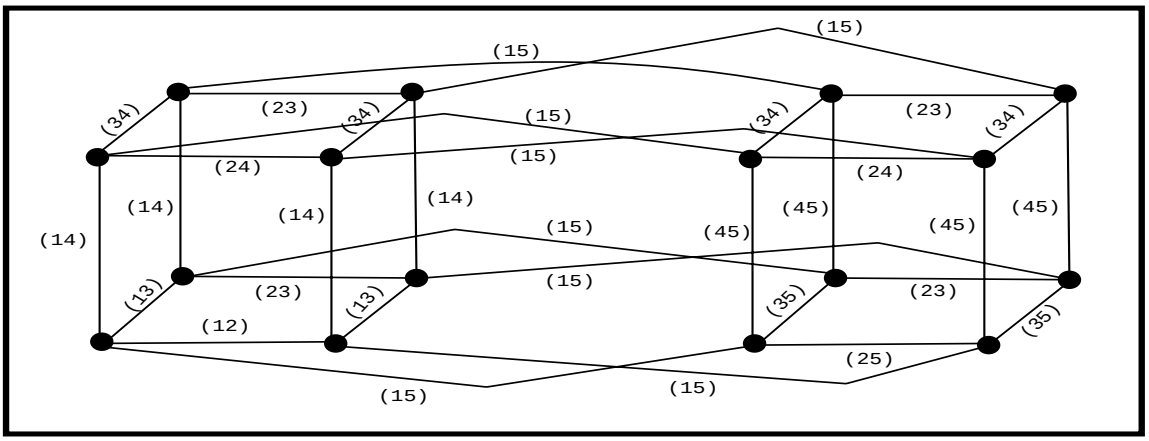
Proof : The proof is by induction. First we show a possible embedding of Q_4 in T_5 (induction base). Then we show that given Q_{n-2} is embeddable in T_{n-1}, it is possible to embed Q_{n-1} in T_n. An embedding of Q_4 in T_5 is depicted in Figure 8(a) where for simplicity only link labels are shown in terms of their corresponding generator actions (i.e. traversal of a link with label (ij) swaps digits in positions i and j). Now suppose the embedding of Q_{n-2} in T_{n-1} is given. Then two copies of of Q_{n-2} can be joined (using two copies of T_{n-1} which are contained in T_n) via a bundle of links labeled as $(1n)$ as shown in Figure 8(b). To find the link labels of the second Q_{n-1}, four possibilities are examined:

Case (i): 3 distinct digits $i \neq j \neq k \neq 1$ are used in labeling a 4-cycle in the first Q_{n-2}. The labeling of the corresponding 4-cycle in the second Q_{n-2} is shown in Figure 9(a).

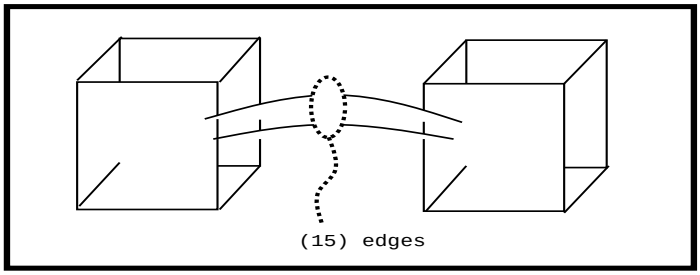
Case (ii): 3 distinct digits $i \neq j \neq k$ and $i = 1$ are used in labeling a 4-cycle. The labeling of the second 4-cycle is shown in Figure 9(b).

Case (iii): 4 distinct digits $i \neq j \neq k \neq \ell \neq 1$ are used in labeling a 4-cycle in the first Q_{n-2}. The labeling of the corresponding 4-cycle in the second Q_{n-2} is shown in Figure 9(c).

Case (iv): 4 distinct digits $i \neq j \neq k \neq \ell$ and $i = 1$ are used in labeling a 4-cycle. The labeling of the second 4-cycle is shown in Figure 9(d).



(a)



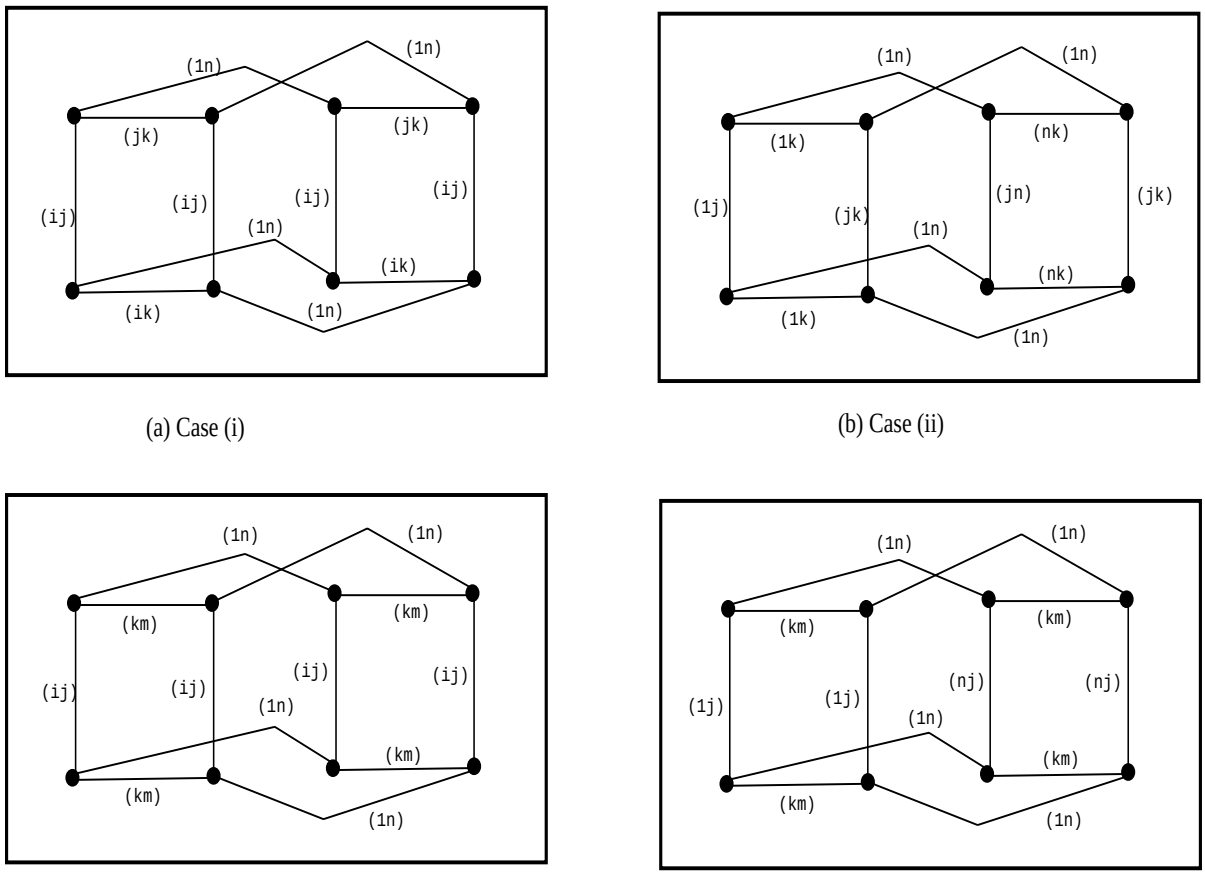
(b)

Figure 8: Embedding of Q_4 in T_5

So, two copies of Q_{n-2} can be joined via links with the label $(1n)$ to create a Q_{n-1}. Since dilation of the first Q_{n-2} was one, the dilation of the new Q_{n-1} will remain one as well. □

Remark 4.6 *The above embedding is in no way optimal; as a matter of fact embedded hypercubes have a very large expansion and the expansion increases with the size of the hypercube. We pose the following question for further investigation. What is the smallest n such that a Q_m can be embedded in T_n? In addition, one may consider packing more than one hypercube in T_n to improve the utilization of available vertices. For example, it can be readily seen from Theorem 4.10 that up to $\lfloor n/2 \rfloor$ Q_{n-1}'s can be embedded in T_n. This is because there exist $\lfloor n/2 \rfloor$ pairs of T_{n-1}'s in T_n, and each pair of T_{n-1}'s contains a Q_{n-1}.*

As discussed previously, no two edge-disjoint S_n's can be embedded in T_n. This can be readily seen from the transposition matrix where each row specifies the set of generators for a given S_n. Since any two such rows, say rows i and j, have the element $g(i,j)$ in common, their corresponding S_n's cannot be edge-disjoint. Now consider a



(a) Case (i)

(b) Case (ii)

(c) Case (iii)

(d) Case (iv)

Figure 9: Four possibilities of embedding Q_{n-1} in T_n

transposition network in which the no. of links in each dimension (or the number of available generators) is doubled. Call such a network a *Doubly linked Transposition Network* or DT_n. A T_n in which half the available bandwidth for each generator action is utilized may also be regarded as a DT_n. Regarding DT_n, the following can be established.

Theorem 4.11 *The doubly-linked Transposition Network DT_n contains n edge-disjoint S_n's.*

Proof : By construction. The generator set of each S_n is specified by one of the rows of the $n \times n$ generator matrix. Since there are two links between two adjacent vertices, $g(i,j)$ is distinct form $g(j,i)$ for any two rows i and j. □

Corollary 4.3 *DT_n can similarly embed n edge-disjoint B_n's.*

Remark 4.7 *Observe that in embedding edge-disjoint S_n's and B_n's in DT_n all the available links have been utilized, and thus the embedding is optimal.*

5 Conclusion

It was shown that the set of generators for any graph isomorphic to n-star is unique up to isomorphism among digits. Furthermore, unlike the hypercube, the star graph was shown to be incapable of achieving a wildcard dimension through adding a single dimension to its existing ones. The Bubble-sort, however, can have one wildcard with only one extra dimension. A systematic method was suggested to add generators in order to obtain a certain number of wildcard generators. The minimum number of generators required for this purpose was derived. For the star graph, the superiority of the proposed method over replication method was for the case of $n = 2k + 1$, where obtaining k wildcard dimensions using the proposed method resulted in k fewer generators as compared to the replication method. Based on the proposed method, a fault tolerant network, known as Transposition graph, was developed which can reconfigure to an n-dimensional star graph despite the failure of up to $(\lceil \frac{n}{2} \rceil - 1)$ links and to an n-dimensional bubble sort graph despite the failure of up to $(n-2)$ links. It is to be noted that these transposition graphs were briefly mentioned in [Lei90, LJD93].

It appears that the different properties like low diameter, strong resilience (vertex and link fault tolerance) enjoyed by the Cayley graphs are dictated by the intrinsic properties of the respective generator sets. The basic nature of the "transposition" operation plays an important role towards contributing to the richness of the star graphs and other families of graphs that use transpositions as their generators. Our purpose in the second part of this paper has been to investigate in details the topological properties of these Transposition Graphs, where the generator set includes all possible transpositions. The transposition graph T_n is regular, has sublogarithmic diameter and a simple shortest routing algorithm and it is optimally fault tolerant (vertex connectivity is $\binom{n}{2}$); it is shown that T_n has wide containers of short lengths (container length is increased only by one over the diameter of the graph). To compare transposition networks with say, hypercubes, T_n has a lesser diameter and wider container; this comes with a price – T_n has a larger node degree. We have also shown how T_n can efficiently embed popular existing architectures like 2D meshes and hypercubes. Obviously, any other Cayley graph of dimension n that uses only transposition

type generators is included in T_n. It'd be interesting to study T_n as a simulator of those graphs under multiple link failure.

Acknowledgements: The authors are grateful to the reviewers for their valued comments that greatly improved the presentation. One of the reviewers brought to our attention the existence of the paper [HLD94] which discusses some aspects of fault tolerance of transposition networks.

References

[AK87a] S. B. Akers and B. Krishnamurthy. The fault tolerance of star graphs. In *Proceedings of 2nd International Conference on Supercomputing*, volume III, pages 270–276, San Francisco, May 1987.

[AK87b] S. B. Akers and B. Krishnamurthy. The star graph: an attractive alternative to n-cube. In *Proceedings of International Conference on Parallel Processing (ICPP-87)*, pages 393–400, St. Charles, Illinois, August 1987.

[AK89] S. B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555–566, April 1989.

[BCH93] J. Bruck, R. Cypher, and C. T. Ho. Wildcard dimensions, coding theory and fault tolerant meshes and hypercubes. In *Proceedings of 23rd International Symposium on Fault Tolerant Computing*, pages 260–267, June 1993.

[DT91] K. Day and A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. Technical Report TR 91–10, Computer Science Department, University of Minnesota, Minneapolis, MN, May 1991.

[DT94] K. Day and A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):31–38, January 1994.

[FA91] P. Fragopoulou and S. G. Akl. Parallel algorithm for computing Fourier transforms on the star graph. In *Proceedings of the International Conference on Parallel Processing*, volume III, pages 100–106, St. Charles, Illinois, 1991.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and INtractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979. pages 199-200.

[HLD94] J. Huang, S. Lakshmivarahan, and S. K. Dhall. Analysis of interconnection networks based on Cayley graphs of strong generating sets. Private Communication, 1994.

[JLD91] J. S. Jwo, S. Lakshmivarahan, and S. K. Dhall. Embedding of cycles and grids in star graphs. *Journal of Circuits, Systems and Computers*, 1(1):43–74, 1991.

[Kav93] A. Kavianpour. System level diagnosis strategies for n-star multiprocessor systems. In *Proceedings of International Conference on Parallel Processing*, pages 297–300, 1993.

[Knu73] D. E. Knuth. *The Art of Computer Programming, Volume III*. Addison-Wesley, 1973.

[Lat91] S. Latifi. Subcube embeddability of folded hypercubes. *Parallel Processing Letters*, 1(1):43–50, 1991.

[Lat93] S. Latifi. On fault diameter of star graphs. *Information Processing Letters*, 46:143–150, June 1993.

[LEA89] S. Latifi and A. El-Amawy. On folded hypercubes. In *Proceedings of International Conference on Parallel Processing*, volume I, pages 180–187, 1989.

[Lei90] F. T. Leighton. *Introductions to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*. Morgan Kaufman, 1990. page 776.

[LJD93] S. Lakshmivarahan, J. S. Jwo, and S. K. Dhall. Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey. *Parallel Computing*, 19:361–407, 1993.

[MP88] F. J. Meyer and D. K. Pradhan. Flip-trees: fault tolerant graphs with wide containers. *IEEE Transactions on Computers*, 37(4):472–478, April 1988.

[MS90] A. Menn and A. K. Somani. An efficient sorting algorithm for the star graph interconnection network. In *Proceedings of the International Conference on Parallel Processing*, volume III, pages 1–8, St. Charles, Illinois, 1990.

[MS92] V. E. Mendia and D. Sarkar. Optimal broadcasting on the star graph. *IEEE Transactions on Parallel and Distributed Systems*, 3(4):389–396, July 1992.

[NSK90] M. Nigam, S. Sahni, and B. Krishnamurthy. Embedding Hamiltonians and hypercubes in star interconnection graphs. In *Proceedings of the International Conference on Parallel Processing*, pages 340–343, August 1990.

[QMA92] K. Qiu, H. Meijer, and S. G. Akl. On the cycle structure of star graphs. Technical Report 92-341, Department of Computer Science, Queen's University, Ontario, Canada, November 1992.

[SS91] S. Sur and P. K. Srimani. Super star: a new optimally fault tolerant network architecture. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS-11)*, pages 590–597, Texas, 1991.

[SS92] S. Sur and P. K. Srimani. Topological properties of star graphs. *Computers & Mathematics with Applications*, 25(12):87–98, 1992.