

Tolerating One and Two Node Failures in a Bipartite Network $K(n,n)$

Abdel Aziz (Zizo) Farrag ⁺

Faculty of Computer Science, Dalhousie University, Halifax, NS, B3H 4R2, Canada

Abstract. To achieve fault tolerance in a parallel or a distributed network, some spare processors and extra links can be added so that the network can continue to operate even in the presence of some failures. However, due to the limitation on the number of links adjacent to a node in a VLSI design, it is important to minimize the node-degree of the whole network. We use this criteria to develop optimal solutions for tolerating one and two node failures in the bipartite graph $K(n,n)$. This configuration includes, as special cases, some important networks (such as stars and trees), and moreover, it was used in modeling many important applications (e.g., matching, scheduling, coding, etc.).

Keywords: fault tolerance, network, graph, bipartite network.

1. Introduction

Fault tolerance is a major concern in the design of parallel and distributed networks [1]. For many applications, it is essential for the network to continue operation even in the presence of faults. To achieve this objective, some spare processors and extra links must be added to the network. In this case, when some of the basic components fail, the network can be reconfigured to bypass and replace these defective components with healthy nodes and links.

Expanding a multicomputer network in order to tolerate failure has been examined for a variety of architectures, (such as rings [2-5], stars [6-8], meshes [9-13], and hypercubes [9, 13, 15]). In this paper, we study this problem for the complete bipartite configuration $K(n,n)$. The optimization criterion used is to reduce the node-degree of the (expanded) network. This is an important objective in practice due to the limitation on the number of links allowed per node in a real implementation [1]. Moreover, this criterion tends to significantly reduce the number of links of the whole network [16].

It is well-known that finding a fault-tolerant design that is optimal (with respect to the above criterion mentioned) is generally very difficult even for networks with a regular structure. Indeed such optimal solutions are known only for a few special cases. We develop optimal solutions for complete bipartite graphs $K(n,n)$ that can tolerate one or two node failures. This generalizes the earlier results [7]. The bipartite configuration is used in many applications, such as scheduling, matching, coding theory, etc. Moreover, several network topologies can be modeled as a bipartite graph such as stars and trees, for examples.

The rest of this paper is organized as follows. The problem is defined in Section 2, and the formulation is given in Section 3. Finding optimal 1-ft and 2-ft solutions of $K(n,n)$ will be examined in Sections 4 and 5; respectively. The 1-ft case was also studied independently in [7]; however, we simplify the proof here to make this paper self-contained. Our conclusions are given in Section 6.

⁺ Corresponding author. Tel.: 902-233-6277.
E-mail address: farrag@cs.dal.ca.

2. The Problem

In what follows, we use a graph to represent a network, where the nodes and the edges correspond to the processors and the links; respectively. The degree of a graph is defined as the maximum degree of any node in it.

Definition 2.1: (graph isomorphism)

A graph G_1 is isomorphic to a graph G_2 (or equivalently, G_1 is an isomorphic copy of G_2) iff there is a 1-to-1 correspondence between their nodes that preserves adjacency, i.e., there is a bijection (function) f from the nodes of G_1 to the nodes of G_2 such that (u, v) is an edge in G_1 iff $(f(u), f(v))$ is an edge in G_2 .

Definition 2.2: (k-ft solution)

A graph H is a k -fault-tolerant (or k -ft) of a graph G iff, for every possible set of k nodes in H , the graph obtained by removing this set of k nodes from H must contain an isomorphic copy of G .

For example, the graph in Fig. 1 has a 1-ft solution shown in Fig. 2. In designing a k -ft solution H of G , we use the minimum number of spare nodes, i.e., H will have exactly k more nodes than G .

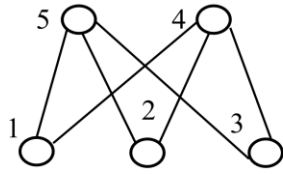


Fig. 1: A graph G .

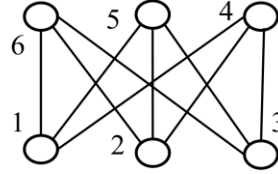


Fig. 2: A 1-ft H of G .

Definition 2.3: (optimal k -ft)

A graph H is called a degree-optimal (or just optimal) k -ft of a graph G , if it is a k -ft of G and has no higher degree than any other k -ft for G .

Definition 2.4: (multipartite graph)

A multipartite (or q -partite) graph is a graph G whose nodes can be divided into q disjoint subsets V_1, V_2, \dots, V_q such that no pair of nodes in the same subset is joined by an edge. If there is an edge in G between every pair of nodes belonging to two different subsets, G is called a complete multipartite graph, and is denoted as $K(|V_1|, |V_2|, \dots, |V_q|)$, where $|V_j|$ is the cardinality of the set V_j , for $1 \leq j \leq q$. A (complete) 2-partite graph is also called a (complete) bipartite graph.

For example, the star graph with $n \geq 2$ nodes is $K(n-1, 1)$. Notice that the complement of $K(|V_1|, |V_2|, \dots, |V_q|)$ consists of the disjoint union of cliques $K_{|V_1|} \cup K_{|V_2|} \cup \dots \cup K_{|V_q|}$. Unless stated explicitly otherwise, we assume the subsets V_1, V_2, \dots, V_q are listed in a non-increasing order, that is, $|V_1| \geq |V_2| \geq \dots \geq |V_q|$.

Given a complete bipartite graph $K(n, n)$, we would like to find an optimal k -ft solution of $K(n, n)$. For arbitrary values of k , this problem appears to be intractable, i.e., may require an exponential search. In this paper, we shall develop optimal 1-ft and 2-ft solutions polynomially, i.e., without performing any search. Except for these cases, no polynomial solution is known for cases $k > 2$.

3. Fault Tolerance of $K(n, n)$

In this section, we develop a characterization for fault tolerance of $K(n, n)$ (based on [7]).

Definition 3.1: (partition)

A partition of n is a collection (or a multiset) of positive integers $\{a_1, a_2, \dots, a_i\}$ whose sum is equal to n . For example, $\{3\}$, $\{2, 1\}$ and $\{1, 1, 1\}$ are all the partitions of 3. The number of elements in the partition is called its length, and the sum of its elements is called its size, e.g., $\{3, 2, 2\}$ is of length 3 and size 7. We do not allow zero elements in any partition, i.e., if an element equals to 0 arises, it will be discarded.

We associate with each graph a partition that represents the sizes of its components, e.g., the graph $K(3, 2)$ shown in Figure 1 has only one component of size 5, i.e., its partition is $\{5\}$, whereas its complement $K^c(3, 2)$ has two components of sizes 3 and 2; respectively, i.e., its partition is $\{3, 2\}$.

Two partitions are considered equal if, after sorting their elements, they will become identical, e.g., $\{2,1\}$ and $\{1,2\}$ are equal. Unless stated explicitly otherwise, we assume the elements of any partition $\{a_1, a_2, \dots, a_i\}$ to be listed in a non-increasing order.

A subpartition of a partition A is one that contains 0, 1, or more of the elements of A , e.g., if $A = \{2,2,1\}$, then $\{2,1\}$ and $\{2,2,1\}$ are examples of subpartitions of A . In what follows, we may loosely use the term subset to mean a subpartition.

Definition 3.2: (refinement of a partition)

We say that a partition A is a refinement of a partition B (or equivalently, A refines B) if A can be obtained by further partitioning zero or more elements of B . For examples, $\{2,2\}$, $\{2,1,1\}$, and $\{1,1,1,1\}$ are all refinements of $\{2,2\}$.

Definition 3.3: (Dominating partition)

A partition $A = \{a_1, a_2, \dots, a_i\}$ is said to dominate a partition $B = \{b_1, b_2, \dots, b_j\}$, (where the elements in both partitions are listed in a non-increasing order), iff the length of A is greater than or equal to the length of B and $a_m \geq b_m$ for all $1 \leq m \leq j$. In this case, we also say B is dominated by A . For example, $\{7,4\}$ dominates $\{7,3\}$, but $\{2,2\}$ does not dominate $\{1,1,1\}$.

Theorem 3.1: Let S be a graph with $2n+k$ nodes, and let the partition of its complement S^c be $\{a_1, a_2, \dots, a_i\}$. Then, S is a k -ft of $K(n,n)$ iff every partition of size $2n$ dominated by $\{a_1, a_2, \dots, a_i\}$ must be a refinement of $\{n,n\}$.

Proof: Assume without loss of generality all components of S^c are complete subgraphs. Suppose first every partition of size $2n$ dominated by $\{a_1, a_2, \dots, a_i\}$ is a refinement of $\{n,n\}$ and let W denotes any $2n$ -node subgraph of S^c , that is, W is obtained by deleting k nodes from S^c . Then, we can group all components of W into two parts each is of size n nodes, i.e., W is a subgraph of $K^c(n,n)$ or equivalently, the complement of W contains $K(n,n)$. This implies S is a k -ft of $K(n,n)$.

Similarly, let B be any partition of size $2n$ dominated by $\{a_1, a_2, \dots, a_i\}$ and is not a refinement of $\{n,n\}$ and let W denotes a subgraph of S^c whose component sizes corresponds to B , i.e., W is obtained by deleting some k nodes from S^c . Then, we cannot group the components of W into two parts each is of size n , i.e., W is not a subgraph of $K^c(n,n)$, or equivalently, the complement of W does not contain $K(n,n)$, which implies S cannot be a k -ft of $K(n,n)$. \square

Example 3.1: By above theorem, $K(2,2,2)$ is a 2-ft of $K(2,2)$, since the only partitions of size 4 dominated by $\{2,2,2\}$ are $\{2,2\}$ and $\{2,1,1\}$; and both are refinements of $\{2,2\}$.

Example 3.2: For any $k \geq 1$, $K(n,n)$ has a k -ft equal to $K(n,n,1,\dots,1)$; i.e., with k 1's at the end.

Example 3.3: $K(4,4)$ is not a 2-ft of $K(3,3)$, since $\{4,2\}$ is dominated by $\{4,4\}$, yet $\{4,2\}$ is not a refinement of $\{3,3\}$.

The proof of the above theorem implies that for any k -ft S of $K(n,n)$, S is either a complete multipartite graph or S contains a complete multipartite graph of size $2n+k$. This proves the following result.

Corollary 3.1: An optimal k -ft of $K(n,n)$ must be a complete multipartite graph. \square

Example 3.4: Both $K(4,4,1,1)$ and $K(4,2,2,2)$ are 2-ft of $K(4,4)$. However, the solution $K(4,2,2,2)$ is cheaper in node degree and (as will be shown later) it is actually optimal.

The above result reduces our search for an optimal k -ft solution considerably, since we only need to check the graphs $K(a_1, a_2, \dots, a_i)$ satisfying the condition of the above theorem and whose least element a_i is the largest (among all other k -ft solutions). We develop optimal solutions for 1-ft and 2-ft solutions in the next sections.

4. Optimal 1-ft of $K(n,n)$

We develop optimal 1-ft for $K(n,n)$. To simplify our results, we present them first in the form of an algorithm which finds the solutions for all values of n . The proofs are given below the algorithm.

```

Algorithm Find_Optimal_1ft (input  $K(n,n)$ ) // Return optimal 1-ft of  $K(n,n)$  {
    if ( $n == 1$ ) return  $K(1,1,1)$ ;
    if ( $n$  is even) return  $K(n/2 + 1, n/2, n/2, n/2)$ ;
    else if ( $3$  divides  $n$ ) return  $K(2n/3 + 1, n/3, n/3, n/3)$ ;
    else if ( $3$  divides  $n-1$ ) return  $K((n+2)/3, (n+2)/3, (n+2)/3, (n-1)/3, (n-1)/3, (n-1)/3)$ ;
    else return  $K((n+4)/3, (n+1)/3, (n+1)/3, (n+1)/3, (n-2)/3, (n-2)/3)$ ;
}

```

It is not hard to check $K(1,1,1)$ is the only 1-ft for $K(1,1)$, and therefore, must be optimal. In the rest of this section, we assume $n \geq 2$. The theorem below identifies optimal solution when n is even.

Theorem 4.1: If n is even, the graph $K(n/2 + 1, n/2, n/2, n/2)$ is an optimal 1-ft of $K(n,n)$.

Proof: Suppose $K(a_1, a_2, \dots, a_i)$ is a 1-ft of $K(n,n)$, and let n be even ≥ 2 . Then, the least element a_i is at most $n/2$. Otherwise, we can obtain a reduced partition of size $2n$ from $\{a_1, a_2, \dots, a_i\}$ by reducing any element whose value is n by 1 or reducing the (least) element a_i by 1, if no element is equal n ; where in both cases, the newly reduced partition will not contain a subset whose size add to n , i.e., cannot be refinement of $\{n,n\}$. This contradicts our assumption that $K(a_1, a_2, \dots, a_i)$ is a 1-ft of $K(n,n)$.

It is not hard to check the graph $K(n/2 + 1, n/2, n/2, n/2)$ is a 1-ft of $K(n,n)$, and since its least element is $n/2$, it must be optimal.

To solve the cases where n is odd, we need first to develop some conditions on the solutions (in the 3 lemmas) below.

Lemma 4.1: Let $K(a_1, a_2, \dots, a_i)$ be a 1-ft of $K(n,n)$, and let its least element a_i be > 1 . Then, the largest element a_1 is $< n$.

Proof: If $a_1 > n+1$ or $a_1 = n$, we can reduce it by 1 to obtain a new partition of size $2n$ that is dominated by $\{a_1, a_2, \dots, a_i\}$ but not a refinement of $\{n,n\}$, as no subset of its elements that includes the reduced element ($a_1 - 1$) will add to n .

Otherwise, if $a_1 = n+1$, we can reduce any other element (e.g., a_2, a_3, \dots , or a_i) by 1 to obtain a new partition of size $2n$ dominated by $\{a_1, a_2, \dots, a_i\}$, but not a refinement of $\{n,n\}$. Thus, all cases above where $a_1 \geq n$ lead to a contradiction.

Lemma 4.2: Let n be odd > 2 and $K(a_1, a_2, \dots, a_i)$ be a 1-ft of $K(n,n)$. Then, the least element a_i is either 1, or the number of elements (i.e., the value of i) must be ≥ 5 .

Proof: It is not hard to check, for example, that $K(n,n,1)$ is a 1-ft of $K(n,n)$. Otherwise, suppose $a_i > 1$ in the 1-ft $K(a_1, a_2, \dots, a_i)$. The proof that the number of the elements (i.e., the value of i) cannot be 1, 2 or 3 follows from lemma 4.1 (which shows each element must be less than n in this case). Otherwise, suppose the number of elements in the partition is 4, i.e., has the form $\{a_1, a_2, a_3, a_4\}$. Then, since the sum of all these 4 elements ($2n+1$) is odd, therefore, either 3 of them are even and one is odd, or 3 of them are odd and one is even. In the former case, we can reduce the odd element by 1 to obtain a new partition of size $2n$ whose elements are all even (and each is $< n$); and in the latter case, we can reduce the only even element by 1 to obtain a new partition whose elements are all odd (and each is $< n$). Thus, in both cases, the reduced partition obtained cannot be a refinement of $\{n,n\}$, which contradicts the assumption that $K(a_1, a_2, \dots, a_i)$ is a 1-ft of $K(n,n)$.

Lemma 4.3: Let n be odd > 2 and $K(a_1, a_2, \dots, a_i)$ be a 1-ft of $K(n,n)$. Then, $a_i \leq n/3$.

Proof: Suppose to the contrary that $a_i > n/3$, this implies that the number of elements (i.e., the value of i) is ≥ 5 (as was shown in the earlier lemma), and since n is odd, this means 3 divides either n , or $n-1$ or $n-2$. If 3 divides n or divides $n-1$, the sum of any 3 elements in the partition $\{a_1, a_2, \dots, a_i\}$ will exceed $n+1$, and therefore, reducing any element in it by 1 will result in a new partition (of size $2n$) that is not a refinement of $\{n,n\}$ since the sum of any 3 elements in it will exceed n . Similarly, if 3 divides $n-2$, the sum of any 3 elements in $\{a_1, a_2, \dots, a_i\}$ will be at least $n+1$, (as a_i will be at least $(n+1)/3$ in this case), and therefore, at least one of the element in this partition must be $> (n+1)/3$, since $2n+1$ is not a multiple of $(n+1)/3$; and therefore, if we reduce the largest element by 1, the resultant partition will not be a refinement of $\{n,n\}$, since the sum

of any 3 or more elements in it will exceed n . Thus, all cases above lead to a contradiction with the assumption that $K(a_1, a_2, \dots, a_i)$ is a 1-ft of $K(n, n)$.

Theorem 4.2: If n is odd integer > 2 , an optimal 1-ft solution of $K(n, n)$ can be defined as follows:

- if 3 divides n , this solution is $K(2n/3 + 1, n/3, n/3, n/3, n/3)$;
- if 3 divides $n-1$, this solution is $K((n+2)/3, (n+2)/3, (n+2)/3, (n-1)/3, (n-1)/3, (n-1)/3)$;
- if 3 divides $n-2$, this solution is $K((n+4)/3, (n+1)/3, (n+1)/3, (n+1)/3, (n-2)/3, (n-2)/3)$.

Proof: It is not hard to check each solution given to be a 1-ft of $K(n, n)$, and moreover by Lemma 4.3, the least element of each solution is the largest possible, i.e., must also be optimal.

Example 4.1: $K(4, 4)$ has an optimal 1-ft equal to $K(3, 2, 2, 2)$.

Example 4.2: $K(15, 15)$ has an optimal 1-ft equal to $K(11, 5, 5, 5, 5)$.

Example 4.3: $K(37, 37)$ has an optimal 1-ft equal to $K(13, 13, 13, 12, 12, 12)$.

5. Optimal 2-ft of $K(n, n)$

We develop optimal 2-ft solutions of $K(n, n)$ in this section. We present our results in the form of an algorithm which finds the solution for all cases of n . The proofs are removed due to the space limitation. (However, the initial draft was reviewed with all the proofs included for all cases.).

Algorithm: Find_Optimal_2ft (input $K(n, n)$) //Return optimal 2-ft of $K(n, n)$ {
 if ($n == 1$) return $K(1, 1, 1, 1)$;
 if ($n == 2$) return $K(2, 2, 2)$;
 if ($n == 3$) return $K(3, 3, 1, 1)$;
 if ($n == 4$) return $K(4, 2, 2, 2)$;
 if ($n > 4$ and 3 divides n) return $K(n/3 + 1, n/3 + 1, n/3, n/3, n/3, n/3)$;
 if ($n > 4$ and 3 divides $n+1$) return $K((n+4)/3, (n+4)/3, (n+1)/3, (n+1)/3, (n-2)/3, (n-2)/3)$;
 if ($n > 4$ and 3 divides $n+2$ and 4 divides n) return $K((n/2, n/2, (n+4)/4, (n+4)/4, n/4, n/4)$;
 if ($n > 4$ and 3 divides $n+2$ and 4 divides $n-2$) return $K((n/2, n/2, (n+6)/4, (n+2)/4, (n+2)/4, (n-2)/4)$;
 if ($n > 4$ and 3 divides $n+2$ and 4 divides $n-1$)
 return $K((n+1)/2, (n+3)/4, (n+3)/4, (n+3)/4, (n-1)/4, (n-1)/4, (n-1)/4)$;
 if ($n > 4$ and 3 divides $n+2$ and 4 divides $n-3$)
 return $K((n-1)/2, (n+5)/4, (n+5)/4, (n+1)/4, (n+1)/4, (n+1)/4, (n-3)/4)$;
}

Example 5.1: $K(4, 4)$ has an optimal 2-ft equal to $K(4, 2, 2, 2)$.

Example 5.2: $K(15, 15)$ has an optimal 2-ft equal to $K(6, 6, 5, 5, 5, 5)$.

Example 5.3: $K(34, 34)$ has an optimal 2-ft $K(17, 17, 10, 9, 9, 8)$.

6. Conclusions

A major concern in designing multicomputer networks is to include fault tolerance capability. This is important for many applications that require the system to operate correctly even in the presence of faults. Fault tolerance is usually achieved by introducing redundancy, i.e., by adding spare processors and extra links. However, due to the limitation on the number of links allowed per node, it is important to minimize the node-degree of the overall network. We have applied this strategy to the design of optimal fault-tolerant solutions of $K(n, n)$ networks. The solutions developed here are optimal and can handle one and two node failures. Beyond this, no polynomial solution is known for solving the problem, that is, there is a lot of room for further research on this configuration.

7. References

- [1] J. Bermond, F. Comellas and D. Hsu, "Distributed Loop Computer Networks: A Survey", J. of Parallel and Dist. Computing, 24, 1995, pp. 2-10.

- [2] Y. Chen, Y. Lin and T. Chen, "Optimal fault-tolerant routing algorithms and fault-tolerant diameter in directed double-loop networks", *Theoretical Computer Science*, V. 468, Jan 2013, pp. 50-58.
- [3] S. Dutt, and J. Hayes, "Designing Fault-Tolerant Systems Using Automorphism", *J. on Parallel and Dist Computing*, 12, no. 3, July 1991, pp. 249-268.
- [4] A. Farrag, "Developing Fault-Tolerant Distributed Loops", *Information Processing Letters*, V. 111, N. 2, Dec 2010, pp. 97-101.
- [5] A. Grnanov, L. Kleinrock, and M. Gerla, "A highly Reliable Distributed Loop Network Architecture", in *Proc. IEEE Symp. Fault Tolerant Computing*, Kyoto, Japan, Oct 1980, pp. 319-324.
- [6] A. Choudum, and S. Sivagurunathan, "Optimal fault-tolerant networks with a server", in *Networks*, V. 35, N. 2, Mar 2000, pp. 157-160.
- [7] A. Farrag, R. Dawson and Y. Qi, "On Designing Fault-Tolerant Extensions with Optimal Fanout for Complete Bipartite Networks", in *Proc. of Parallel Architecture and Languages, Lecture Notes in Computer Science*, Springer-Verlag, June 1992, Paris, France, pp. 177-191.
- [8] E. Schmitter and P. Baues, "The Basic Fault-Tolerant System", *IEEE Micro*, 4, no. 1, February 1984, pp. 66-74.
- [9] J. Bruck, R. Cypher and C. Ho, "Fault-Tolerant Meshes and Hypercubes with Minimal Number of Spares", *IEEE Trans on Comp*, 42, no. 9, September 1992, pp. 1089-1103.
- [10] Y. Chuang, L. Hsu and C. Chang, "Optimal 1-edge Fault-Tolerant Designs for Ladders", *Info. Proc. Letters*, 84, no. 2, October 2002, pp. 87-92.
- [11] F. Chung, F. Leighton, and A. Rosenberg, "Diogenes: a Methodology for Designing Fault-Tolerant VLSI Processor Arrays", in *Proc. IEEE 13th Conf on Fault Tolerant Computing Symp*, June 1993, Chicago, IL, pp. 26-32.
- [12] A. Farrag and S. Lou, "Applying Fault-Tolerant Solutions of Circulant Graphs to Multi-Dimensional Meshes" in *Computers & Mathematics Journal*, 50, no. 8-9, November 2005, pp. 1383-1394.
- [13] L. Snyder, "Introduction to the Configurable, Highly Parallel Computer," *IEEE Computer*, 15, no. 1, January 1992, pp. 47-56.
- [14] A. Farrag, S. Lou and Y. Qi, "Fault-tolerance and Reconfiguration of Circulant Graphs and Hypercubes", in *Proc. of High Performance Computing Symposium (HPCS-08)*, Ottawa, Ontario, Canada, Apr 2008, pp. 475-481.
- [15] D. Rennels, "On Implementing Fault-Tolerance in Binary Hypercubes", *Digest of papers of IEEE Symp on Fault-Tolerant Comp*, July 1986, Vienna, Austria, 344-349.
- [16] J. Hayes, "A Graph Model for Fault-Tolerant Computing Systems," *IEEE Trans. on Computers*, V. c-25, N. 9, September 1976, pp. 875-884.