Optimal Adaptive Fault Diagnosis for Simple Multiprocessor Systems

Evangelos Kranakis[†‡] Andrzej Pelc[*‡] Anthony Spatharis[†]

Abstract

We study *adaptive system-level fault diagnosis* for multiprocessor systems. Processors can test each other and future tests can be selected on the basis of previous test results. Fault-free testers give always correct test results, while faulty testers are completely unreliable. The aim of diagnosis is to determine correctly the fault status of all processors. We present adaptive diagnosis algorithms for systems modeled by *trees*, *rings* and *tori*. These algorithms use the smallest possible number of tests in each case. Our results also imply optimal diagnosis for more general systems, assuming a small number of faults. The cost of adaptive diagnosis turns out to be significantly smaller than that of classical (*one-step*) diagnosis.

[*]Départment d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada. E-mail: pelc@uqah.uquebec.ca.

[†]School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada. E-mail: {kranakis, spathari}@scs.carleton.ca.

[‡]Part of this research was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

1 Introduction

Multiprocessor systems are vulnerable to faults and fault-tolerant computing has become very important for systems with a large number of processors linked together by an interconnection network. The increasing complexity of multiprocessor systems has triggered intense research on problems of *fault detection, diagnosis* and *reconfiguration.* In particular, system-level diagnosis is an important technique in improving the system reliability and availability.

We use the original and widely studied *PMC* model, which was introduced in the classic paper by Preparata, Metze and Chien [11] as the basic framework for system-level diagnosis. In this model, a multiprocessor system is represented by an undirected graph $G = (V, E)$ in which vertices (nodes) in V represent processors (also called *units* or *processing elements*) and edges in E represent communication links. Each node is either *fault free* (denoted by +) or *faulty* (-). Further, we consider a directed weighted testing graph $G^* = (V, E^*)$ (called a *test assignment*) in which $(v_i, v_j) \in E^*$, only if $\{v_i, v_j\} \in E$. The unit v_i (a *tester*) is said to test unit v_j (*tested unit* or *testee*). The above restriction indicates that tests can be conducted only along links of the system. For a test $(v_i, v_j) \in E^*$, the binary edge weight $w(v_i, v_j)$ is 0 (respectively, 1) if v_i evaluates v_j as fault free (respectively, faulty). If $w(v_i, v_j) = 0$ (1), then we say that v_i has a 0-*edge* (respectively, 1-*edge*) to v_j. The collection of all test results is called a *syndrome.*

Three assumptions are made in the *PMC* model.

1. Faults are permanent.
2. Results of tests performed by fault-free processors are always accurate.
3. Results of tests performed by faulty processors are unreliable.

More specifically, the weight function describing test results must satisfy the following condition:

$$w(v_i, v_j) = \begin{cases} 0 & \text{if } v_i \text{ is fault free and } v_j \text{ is fault free} \\ 1 & \text{if } v_i \text{ is fault free and } v_j \text{ is faulty} \\ 0 \text{ or } 1 & \text{otherwise} \end{cases}$$

Many variations of this model of system-level diagnosis have been studied, and significant efficient algorithms and related complexity results have been proposed [1, 3, 7, 9, 10]. Researchers investigated both worst-case location of faults [2, 4, 9] and their random distribution [3, 10].

The aim of fault diagnosis is to locate all faulty units in the system, based on the syndrome obtained. A system with a *testing graph* $G^* = (V, E^*)$ is *one-step t-diagnosable* (or simply *t-diagnosable*), if all faulty units in the system can always be identified, provided that the number of faults does not exceed t and all tests are selected in advance. We require that diagnosis be correct and complete; that is, all faulty units must be diagnosed as faulty and all fault-free units as faulty free. A diagnosis algorithm working for any configuration of at most t faults is called *one-step t-diagnosis.*

Preparata et al. [11] studied test assignments which provide sufficient information to perform correct and complete one-step t-diagnosis. They proved that if as many as t processors may be faulty, then it is necessary for a system to contain n units with $n \geq 2t+1$ in order to be t-diagnosable. Moreover, it is necessary for each unit to be tested by at least t distinct other units. Hakimi and Amin [4] gave a characterization of t-diagnosable systems.

Nakajima [8] first introduced *adaptive diagnosis*, where future tests can be determined dynamically, on the basis of previous test results. The important aspect of this method is to identify a fault-free processor as soon as possible, and then use it to diagnose other units in the system. Hakimi and Nakajima [5] proposed an adaptive diagnosis algorithm for complete graphs taking at most $|V| + 2t - 2$ tests to identify all faulty processors, assuming that the number of faults is at most t.

Blecher [2] and Wu [12] improved the previous bound to $|V|+t-1$. Moreover, Blecher showed that this is also the lower bound on the number of tests to identify all faulty units in the worst case. However, these results assume that the underlying system is modeled by a complete graph. For larger number of nodes, completely connected systems are difficult to implement, which significantly restricts the applicability of the above diagnosis.

We are interested in t-fault adaptive diagnosis algorithms that minimize the number of tests for multiprocessor systems modeled by *trees*, *rings* and *tori*, provided that less than half of the total number of processors are faulty and t does not exceed the minimum degree d of the nodes in the system. These are necessary conditions to perform diagnosis, in view of [11]. Our main results are adaptive diagnosis algorithms using the minimum number of tests in the worst case, for all these systems. This is for the first time that optimal adaptive diagnosis is given for these important architectures. In a forthcoming paper [6], we give very efficient adaptive diagnosis for hypercubes.

Our result for trees implies that adaptive diagnosis of at most one fault, using n tests in the worst case, is possible for any system represented by a connected graph. On the other hand, the result for rings implies adaptive diagnosis of at most 2 faults using $n+1$ tests, for any system represented by a Hamiltonian graph.

This paper is organized as follows. In Sections 2, 3 and 4, we describe optimal adaptive diagnosis algorithms for trees, rings and tori, respectively. Section 5 contains conclusions.

2 Adaptive diagnosis in trees

In this section, we describe an adaptive diagnosis algorithm that minimizes the number of tests for a tree $\mathcal{T}_n$ with n nodes. Since every tree has nodes of degree 1 (leaves), we have to assume that the maximum number of faults is $t = 1$. We also need to assume that $n \geq 2t+1 = 3$. In view of [2, 12], the worst-case lower bound on the number of tests to diagnose at most one fault in any n-node graph, is n. Our algorithm uses n tests for the tree $\mathcal{T}_n$ in the worst case, which is thus optimal.

The following is a description of our adaptive testing procedure for the tree $\mathcal{T}_n$. We will show that it can be used to perform correct diagnosis.

Procedure Ada-TreeTest$(\mathcal{T}_n)$

Step 1:
Choose any node of degree ≥ 2 as a root of the tree. Perform the first series of tests along all edges of the tree, from parent to child.

Step 2:
If (there are no 1-edges in the syndrome **or** there is exactly one 1-edge which starts at the root r) **then** perform one additional test $(s,\ r)$, where $(r,\ s)$ was a 0-edge in Step 1.

Theorem 2.1 *A tree $\mathcal{T}_n$ with n nodes, $n \geq 3$, can be correctly diagnosed using procedure* Ada-TreeTest, *provided that the number of faulty units is at most 1. The worst-case number of tests is n.*

Proof: Consider the first series of $n-1$ tests (*Step* 1) and let k be the maximum number of consecutive 1-edges. We consider the following fault cases.

Case 1: $k \geq 3$. This is impossible, because at least one end-node of every 1-edge must be faulty and we have $t = 1$.

Case 2: $k = 2$. The unique faulty node has to be the common end-node of the two consecutive 1-edges. All other nodes must be fault-free.

Case 3: $k = 1$. If there are more than one 1-edges in Step 1, they must have a common beginning u, because otherwise the number of faults would exceed 1. In this case, u is the unique faulty node. (This must be the root, otherwise the parent of u would be faulty as well.) If there is only one 1-edge (u, v) in Step 1, there are two possibilities:

3.1. u has a parent u'. Then both u and u' must be fault free because (u', u) is a 0-edge. In this case v is the unique faulty node.

3.2. u is the root. Let v be its child for which (u, v) is a 1-edge. Let v' be a child of u different from v. Hence (u, v') is a 0-edge. v' must be fault free (otherwise there would be at least two faults). Perform one last test (v', u). If its outcome is 1, then u is the unique faulty node. If its outcome is 0, then u is fault free and v is the unique faulty node.

Case 4: $k = 0$. If the root r is fault-free, then all other nodes must be fault-free, by the properties of the model. If r is faulty, then it is the unique faulty node. We conclude that all other nodes must always be fault-free. Thus, we perform one additional test $((v, r)$, where v is any child of r. If the outcome is 0, then all nodes are fault free. If the outcome is 1, then r is the only faulty node.

It follows that n tests are always sufficient to correctly diagnose the status of all faulty units in any tree $\mathcal{T}_n$, if $n \geq 3$ and $t \leq 1$. ∎

We conclude this section with the following proposition which shows that adaptive diagnosis is more efficient than one-step diagnosis in the case of trees.

Proposition 2.2 *The minimum number of tests, sufficient to perform one-step diagnosis of at most 1 fault in any tree with $n \geq 3$ nodes, is $n+1$.*

Proof: In order to prove the upper bound, we need to show a test assignment using $n+1$ tests. Let r be any node of degree at least 2 in the tree T. Consider T as rooted at r and let a and b be two children of r. The test assignment consists of all tests from parent to child plus two additional tests (a, r) and (b, r). A case by case analysis similar to that of the adaptive algorithm shows that any syndrome for this test assignment can be diagnosed provided that there is at most one fault.

In order to prove the lower bound, first observe that each node has to be tested by at least one neighbor. We prove the lower bound by induction on n. For $n = 3$ the tree has two leaves A and C and a node B of degree 2. If only 3 tests were performed, the node B would have only one tester, say A. In this case the syndrome giving outcome 1 to both tests on link AB and 0 to the test (B, C) cannot be diagnosed, as two fault patterns are possible: the unique fault is either A or B.

Suppose the proposition is true for n. Consider any tree T with $n+1$ nodes. Let u be any leaf of T and v its unique neighbor. If there is no test (u, v) in the test assignment then the tree $T' = T \setminus \{u\}$ must be diagnosed separately, without using u. By the inductive hypothesis this requires at least $n+1$ tests in T'. Together with the obligatory test (v, u) this gives at least $n+2$ tests.

It remains to consider the case when the test (u, v) is in the test assignment. We will show that v must have another tester, apart from u. Otherwise, the syndrome giving outcome 1 for both tests along link uv and 0 for all other tests cannot be diagnosed, as two fault patterns are possible: the unique fault is either u or v. Hence we conclude that all nodes of T other than v have at least one tester and v has at least two testers, yielding a total of at least $n+2$ tests. ■

3 Adaptive diagnosis in rings

In this section we describe an adaptive diagnosis algorithm which determines the status of all processors for ring systems in an optimal number of tests. Since the degree of each node is 2, we have to assume that the maximum number of faults is $t = 2$. We also need to assume that $n \geq 2t+1 = 5$. By the result of [2, 12], the lower bound on the number of tests is $n+1$, even in the complete graph $\mathcal{K}_n$. Our algorithm uses $n+1$ tests for the n-node ring in the worst-case, which is thus optimal. The following is a description of our adaptive testing procedure. We will show that it can be used to perform correct diagnosis.

Procedure Ada-RingTest(n)

Step 1:
Perform the first series of tests along all edges of the ring in the clockwise direction.

Step 2:
If there is a sequence $a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{1} d \xrightarrow{0} e$ in the syndrome from *Step* 1
then perform one additional test (e, d);

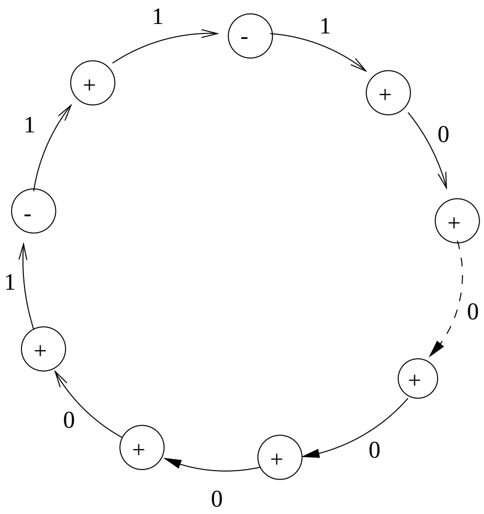

Figure 1: Unique Fault Pattern in Rings for $k = 4$.

If there is a sequence $a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{0} d$ and there are only two 1-edges in the syndrome from *Step* 1 **then** perform one additional test (d, c);

If there is a sequence $a \xrightarrow{1} b \xrightarrow{0} c \xrightarrow{1} d \xrightarrow{0} e$ and there are two 1-edges in the syndrome from *Step* 1 **then** perform one additional test (e, d);

If there is a sequence $a \xrightarrow{1} b \xrightarrow{0} c \xrightarrow{0} d$ and there is one 1-edge in the syndrome from *Step* 1 **then** perform one additional test (d, c).

Theorem 3.1 *A ring $\mathcal{R}_n$ with $n \geq 5$ nodes can be correctly diagnosed using procedure* Ada-RingTest*, provided that the number of faults is at most* 2*. The worst-case number of tests is* $n + 1$*.*

Proof: Let k be the maximum number of consecutive 1-edges in Step 1. We consider the following cases.

Case 1: $k \geq 5$. This is impossible, because at least one end-node of every 1-edge must be faulty and we have $t = 2$.

Case 2: $k = 4$. Since every 1-edge must be incident to a faulty node and $n \geq 5$, there is a unique possible fault pattern in this case for the ring system as it is shown in Fig. 2. The total number of tests is n.

Case 3: $k = 3$. In this case there can be no other 1-edges. Hence we must have the following part of the syndrome $v_1 \xrightarrow{0} v_2 \xrightarrow{1} v_3 \xrightarrow{1} v_4 \xrightarrow{1} v_5 \xrightarrow{0} v_6$ (with v_1 possibly equal to v_6). Since all faulty nodes must be incident to 1-edges, we conclude that v_1 and v_6 must be fault free. Hence v_2 has to be fault free as well. Consequently, v_3 must be faulty. One and exactly one of v_4 and v_5 has to be

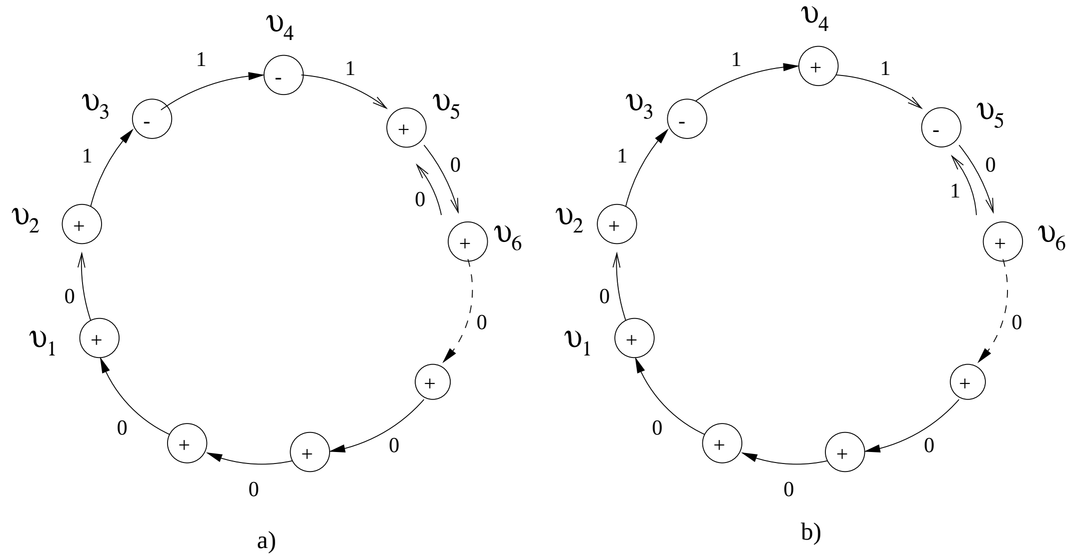

Figure 2: Adaptive Diagnosis in Rings for $k = 3$.

faulty. We perform one extra test (v_6, v_5). If the result is 0, then v_5 is fault free and thus the only faulty nodes are v_3, v_4 (see Fig. 3 a). If the result is 1, v_5 is faulty and thus the only faulty nodes are v_3, v_5 (see Fig. 3 b). Thus, the total number of tests is $n + 1$.

Case 4: $k = 2$. There can be at most two non-consecutive 1-edges, otherwise the number of faulty nodes would exceed 2. We consider three possibilities:

4.1. There are two sequences $\bigcirc \xrightarrow{1} \bigcirc \xrightarrow{1} \bigcirc$ in the syndrome. In this case, the only two faulty nodes are the middle nodes of these sequences. The total number of tests is n.

4.2. There is one sequence $\bigcirc \xrightarrow{1} x \xrightarrow{1} \bigcirc$ and the other one is the 1-edge $u \xrightarrow{1} v$. In this case, one of the faulty nodes is the middle node x of the sequence and the other must be either u or v. Consider the tester z of the unit u from Step 1. The result of test (z, u) is 0 and z must be fault free. Hence u must be fault free and v must be faulty. The only faulty nodes in this case are x and v. The total number of tests is n.

4.3. There is one sequence $u \xrightarrow{1} v \xrightarrow{1} w$ and all other edges are 0-edges. Consider the following part of the syndrome: $x \xrightarrow{0} u \xrightarrow{1} v \xrightarrow{1} w \xrightarrow{0} y$. If u were faulty, then x would have to be faulty as well. Thus, the number of faulty units would exceed two because one of v and w must also be faulty. Hence v is faulty. If node y were faulty, then w would be faulty as well and the number of faulty nodes would exceed 2. Thus, y has to be fault free. We perform one additional test (y, w). If the result of this test is 0, then w is fault free and the only faulty unit is v. If the result of this test is 1, then w is faulty, and the only faulty units are v and w in this case. The total number of tests is $n + 1$.

Case 5: $k = 1$. We consider the following two possibilities:

5.1. There is one 1-edge in the syndrome. Consider the following part of the syndrome in the pattern $x \xrightarrow{0} y \xrightarrow{0} u \xrightarrow{1} v \xrightarrow{0} z \xrightarrow{0} t$. If the node y were faulty, then x would also have to be faulty. Thus, the number of faulty nodes would exceed 2, as either u or v must be faulty as well. Hence y must be fault free, u must be fault free and v must be faulty. Also, node t (and all nodes following t and preceding x in the clockwise direction, cf. Fig. 4 a) have to be fault free for the same reason as node y. Perform one extra test (t, z). If the outcome is 0, z must be fault free and the only faulty node is v in this case. If the outcome is 1, then z is faulty and the only faulty nodes are v and z. The total number of tests is $n + 1$.

5.2. There are two 1-edges in the syndrome.

5.2.1. Each of the 1-edges is followed by at least two consecutive 0-edges. An argument similar to that in Case 4.2 shows that the only possible fault pattern is the one shown in Fig. 4 b. The total number of tests is n.

5.2.2. One of the 1-edges (y, z) is followed by a 0-edge $(z,\ t)$ followed by the other 1-edge (t, u) (see Fig. 4 c). Since $n \geq 5$, the tester x of y must be distinct from u. Let w be the clockwise neighbor of u (possibly $w = x$). A similar argument as before shows that all nodes between w and y (clockwise, with w and y included) have to be fault free. Hence, z must be faulty. Exactly one of nodes t or u must be faulty. Perform one extra test (w, u). If its result is 0, then u is fault free and t is faulty. The only faulty nodes in this case are z and t. If its result is 1, u is faulty and t is fault free. The only faulty nodes in this case are u and z. The total number of tests is $n + 1$.

Case 6: $k = 0$. The existence of any faulty unit would imply that all processors are faulty, which contradicts the upper bound 2. Hence all units must be fault free. The total number of tests is n in this case. ∎

The following proposition shows that the efficiency gain due to adaptive way of testing in the case of rings is even larger than for trees.

Proposition 3.2 *The minimum number of tests sufficient to perform one-step diagnosis of at most 2 faults in any ring with $n \geq 5$ nodes, is $2n$.*

Proof: Straightforward. ∎

4 Adaptive diagnosis in tori

In this section we give optimal adaptive diagnosis algorithms for systems whose underlying graph is a torus. A $k \times m$ -torus (also called a *wrap-around mesh*) is a graph product of cycles of length k and m (considering an isolated node as a cycle of length 1 and a segment as a cycle of length 2). That is, a $1 \times m$ -torus is a cycle of length m and a 2×2 -torus is a cycle of length 4. A $2 \times m$ -torus, for $m \geq 3$ has degree 3, and a $k \times m$ -torus, for $k, m \geq 3$ has degree 4. Therefore,

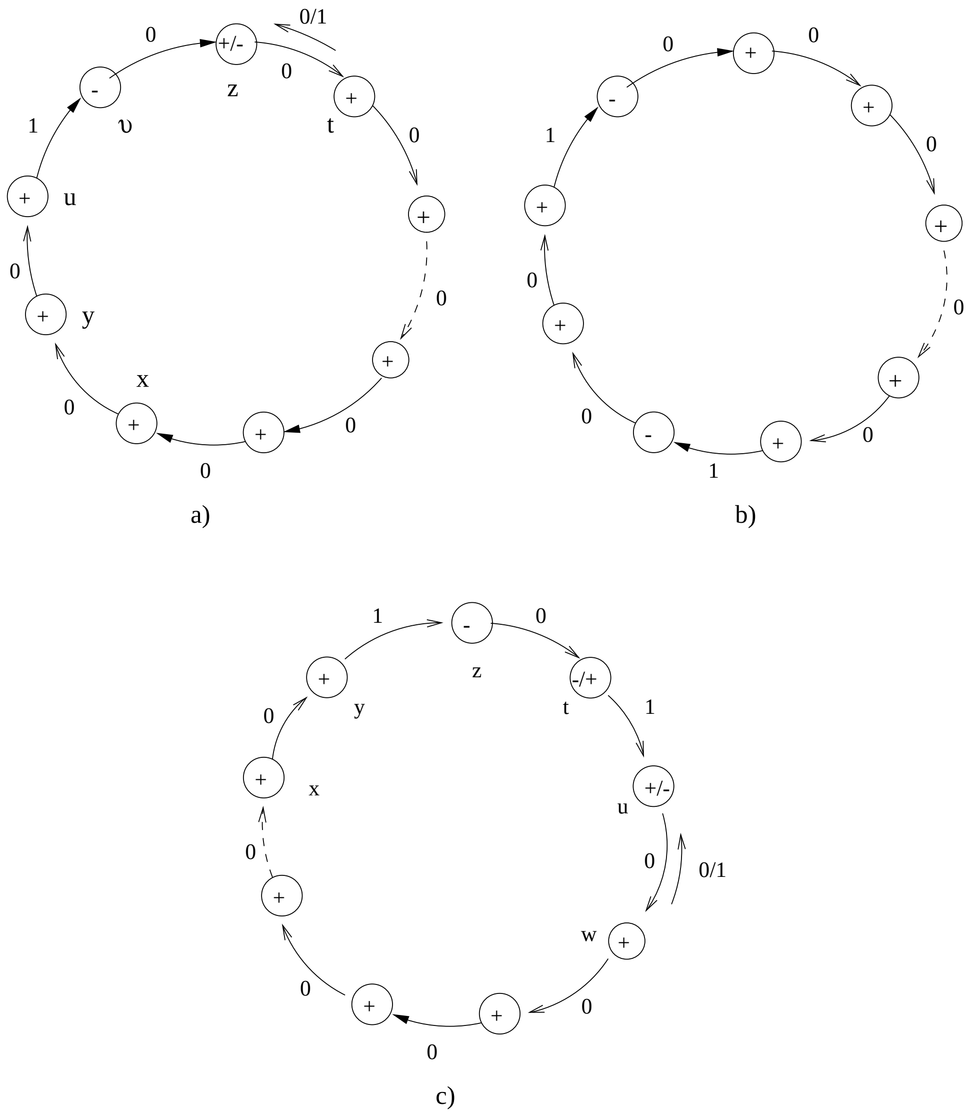

Figure 3: Adaptive Diagnosis in Rings for $k = 1$.

the maximum number of faults that can be diagnosed in a $2 \times m$ -torus, is 3, and the maximum number of faults that can be diagnosed in a $k \times m$ -torus, for $k, m \geq 3$, is 4.

4.1 Diagnosis of a $2 \times m$ -torus

Since we want to diagnose any configuration of at most 3 faults, we need to assume that the number of nodes $2m$ exceeds 6, and hence $m \geq 4$. Consider the two disjoint rings of length m in the torus and call one of them *first* and the other *second*. Fix the same direction in both rings and call it *clockwise*. We will show an adaptive diagnosis algorithm that diagnoses any configuration of at most 3 faults in a $2 \times m$ -torus, for $m \geq 4$, using at most $2m + 2$ tests in the worst case. This is optimal, according to the lower bound from [2]. Indeed, the lower bound on the number of tests to diagnose at most 3 faults in any n-node graph, is $n + 3 - 1$ which equals $2m + 2$ in our case.

First assume that $m \geq 5$. We use the following testing procedure.

Procedure Ada-TorusTest $2 \times m$

Step 1:
Perform consecutive tests $(v_1, v_2), ..., (v_{k-1}, v_k)$ along edges of the first ring in the clockwise direction, until a result 1 is obtained or all tests are done.

Step 2:
If all results are 0 {Case 1}
then test every node in the second ring by its neighbor in the first ring;

If m tests were performed and the last result was 1 {Case 2}
then
    complete testing in the second ring, as in *Ada-RingTest*,
    assuming that the number of faults is at most 2;

If less than m tests were performed {Case 3}
then
    perform all tests along edges of the second ring in the clockwise direction.
    If all results are 0 {Case 3.1}
    then
        test nodes $v_1, v_2, ...$ in the first ring by their neighbors
        in the second ring until a result 0 is obtained;
        If less than 3 tests were done {Case 3.1.1}
        then
            test all nodes in the first ring, other than
            $v_1, v_2, ..., v_k$, by their neighbors in the second ring
    else {Case 3.2}
        complete testing in each ring separately, as in *Ada-RingTest*,
        assuming that the number of faults in each ring is at most 2.

Theorem 4.1 *A $2\times m$ -torus, for $m \geq 5$, can be correctly diagnosed using procedure* Ada-TorusTest $2 \times m$*, provided that the number of faults is at most 3. The worst-case number of tests is $2m + 2$.*

Proof:
Case 1: All test results in Step 1 are 0.
All nodes in the first ring are fault free. Hence these nodes can be trusted and all nodes in the second ring are diagnosed in Step 2. $2m$ tests are used in this case.

Case 2: m tests were performed in Step 1 and the last result was 1.
There is exactly one fault in the first ring and the faulty node is the node v for which test (u, v) gave result 1. Now there are at most 2 faults in the second ring and they can be diagnosed using at most $m + 1$ tests, as in section 3. The total worst-case number of tests is $2m + 1$.

Case 3: $i < m$ tests were performed in Step 1.
3.1. m results 0 were obtained when testing the second ring.
All nodes in this ring are fault free. They start testing neighbors in the first ring until a result 0 is obtained. Every result 1 discovers a faulty node. A result 0 while testing node v_r, for $r < k$, permits to diagnose $v_r, ..., v_{k-1}$ as fault free and v_k as faulty. In this way, all nodes $v_1, v_2, ..., v_k$ are diagnosed.

3.1.1: Less than 3 tests were done.
Testing all other nodes in the first ring by their neighbors in the second ring will diagnose all remaining nodes using a total of at most $2m + 2$ tests.

3.1.2: 3 tests were done.
3 faults were discovered and there is no need of further testing. The total number of tests in this case is $i + m + 3 \leq 2m + 2$.

3.2.: Some results 1 were obtained when testing the second ring.
There must be faults in each of the two rings. Consequently, there are at most two faults in each of them, and hence each of them can be diagnosed using at most $m + 1$ tests, as in section 3. The total worst-case number of tests is $2m + 2$. ■

The case of the 2×4 -torus is similar to the above, with the exception that 2 faults are not always possible to diagnose in a ring of length 4, without using testers from outside. However, in this case as well, diagnosis can always be done using (the optimal number of) 10 tests. We omit the details.

Corollary 4.2 *A $2 \times m$ -torus, for $m \geq 4$, can be correctly diagnosed using at most $2m + 2$ tests in the worst case, provided that the number of faults is at most 3. This number of tests is optimal.*

4.2 Diagnosis of a $k \times m$ -torus, for $k, m \geq 3$

Since the degree of each node is 4, we assume that the number of faults is at most 4. We will show an adaptive diagnosis algorithm that diagnoses any configuration of at most 4 faults in a $k \times m$

-torus, for $k \geq 3$ and $m \geq 4$, using at most $km+3$ tests in the worst case. This is optimal, according to the lower bound from [2]. Indeed, the lower bound on the number of tests to diagnose at most 4 faults in any n-node graph, is $n+4-1$ which equals $km+3$ in our case.

First consider the case when $m \geq 5$.

We call the k rings of length m *horizontal.* Label consecutive horizontal rings with integers 0 to $k-1$. All operations on labels are modulo k. For each horizontal ring specify one direction called *clockwise.* We will use phrases "ring s tests ring $s-1$" and "ring s tests ring $s+1$" to describe the sequence of m tests by nodes of ring s on their neighbors in the respective ring. We will use two subroutines, whose input is the label i of a horizontal ring.

Subroutine PartialClockwise(i)

Perform consecutive tests $(v_1^i, v_2^i), ..., (v_{r-1}^i, v_r^i)$ along edges of ring i in the clockwise direction
until a result 1 is obtained or all tests are done;
mark ring i "tested".

Subroutine CascadeTestRight(i)

$j := i$;
  ring j tests ring $j+1$;
  mark ring $j+1$ "tested";
  $j := j+1$
until some test results are 1 or ring $j+1$ "tested".

The subroutine *CascadeTestLeft*(i) is defined similarly, by replacing $j+1$ by $j-1$ in the above. If ring i consists of fault-free nodes, all test results obtained during subroutines *CascadeTestRight*(i) and *CascadeTestLeft*(i) can be trusted.

The last subroutine uses as input nodes $v_1^i, ..., v_r^i$ generated by *PartialClockwise*(i).

Subroutine RightNeighborTest(i)

test nodes $v_1^i, v_2^i, ...$ in ring i by their neighbors in ring $i+1$
until a result 0 is obtained;
If less than 4 tests were done
then
  test all nodes in ring i, other than
  $v_1^i, ..., v_r^i$, by their neighbors in ring $i+1$.

The subroutine *LeftNeighborTest*(i) is defined similarly, replacing $i+1$ by $i-1$ in the above. If all nodes in the testing ring are fault free, these subroutines permit to diagnose all nodes in the tested ring, similarly as in Case 3.1. in the proof of Theorem 4.1.

We will often perform all tests in the clockwise direction on a horizontal ring. If all test results are 0, the ring will be called *healthy*, otherwise it will be called *unhealthy.* Since $m \geq 5$ and the number of faults is at most 4, a healthy ring cannot contain any faulty nodes.

We define the following testing procedure working for a $k \times m$ -torus, for $k \geq 3$ and $m \geq 5$, and using the above described subroutines.

Procedure Ada-TorusTest-General $(k \times m)$

```
PartialClockwise(0);
If all results in PartialClockwise(0) are 0 {Case 1}
then
    CascadeTestRight(0);
    CascadeTestLeft(0);
    apply Ada-RingTest to each non "tested" ring separately,
    assuming that the number of faults in each ring is at most 2;
else
    PartialClockwise(1);
    If all results in PartialClockwise(1) are 0 {Case 2}
    then
        CascadeTestRight(1);
        If less than m tests were performed during PartialClockwise(0)
        then RightNeighborTest(0);
        apply Ada-RingTest to each non "tested" ring separately,
        assuming that the number of faults in each ring is at most 2;
    else
        PartialClockwise(2);
        If all results in PartialClockwise(2) are 0 {Case 3}
        then
            CascadeTestRight(2);
            If all test results in CascadeTestRight(2) are 0 {Case 3.1}
            then
                If less than m tests were performed during PartialClockwise(1)
                then RightNeighborTest(1);
                If less than m tests were performed during PartialClockwise(0)
                then LeftNeighborTest(0);
            else {Case 3.2}
                complete testing in each ring separately, as in Ada-RingTest,
                assuming that the number of faults in each ring is at most 2
        else {Case 4}
            complete testing in each ring separately, as in Ada-RingTest,
            assuming that the number of faults in each ring is at most 2.
```

Theorem 4.3 *A $k \times m$ -torus, for $k \geq 3$ and $m \geq 5$, can be correctly diagnosed using procedure* Ada-TorusTest-General, *provided that the number of faults is at most 4. The worst-case number of tests is $km + 3$.*

Proof:
Case 1: Ring 0 is healthy.

Subroutines *CascadeTestRight*(0) and *CascadeTestLeft*(0) either diagnose all horizontal rings or diagnose two rings containing faults. Hence each non diagnosed horizontal ring contains at most two faults and *Ada-RingTest* can be applied to these rings. At most two of those rings are unhealthy, hence the total worst-case number of tests is $km+2$.

Case 2: Ring 0 contains faulty nodes and ring 1 is healthy.

Subroutine *CascadeTestRight*(1) either (2.1) diagnoses all horizontal rings other than 0 or (2.2) diagnoses another ring containing faults.

2.1. If m tests were performed during *PartialClockwise*(0), there is at most 1 fault in ring 0 and it can be discovered without further testing. A total of km tests were used. If $i < m$ tests were performed during *PartialClockwise*(0), a similar argument as that in Case 3.1. in the proof of Theorem 4.1 shows that at most $m+3$ tests are used to diagnose ring 0 by *PartialClockwise*(0) together with *RightNeighborTest*(0), for a total of at most $km+3$ tests.

2.2. Ring 0 contains at most 3 faults. If there are 3 faults in this ring, they will be diagnosed by *PartialClockwise*(0) together with *RightNeighborTest*(0) using at most $m+3$ tests. In this case there are no faults in the non diagnosed rings. The total number of tests is $km+3$.

If there are at most two faults in ring 0 they will be diagnosed using $m+2$ tests, similarly as before. The j non diagnosed rings contain a total of at most 2 faults, and applying *Ada-RingTest* to each of them separately permits to diagnose them using at most $jm+1$ tests, for a total of at most $km+3$ tests.

Case 3: Rings 0 and 1 contain faulty nodes and ring 2 is healthy.

3.1. All test results in *CascadeTestRight*(2) are 0. Hence the only rings containing faulty nodes are rings 0 and 1. If m tests were performed during *PartialClockwise*(0) or *PartialClockwise*(1) then the respective ring contains exactly one fault and it can be discovered without further testing. A similar argument as before shows that both rings can be diagnosed by *LeftNeigborTest*(0) and *RightNeigborTest*(1), respectively, and the total number of tests in the entire procedure does not exceed $km+3$.

3.2. *CascadeTestRight*(2) discovered faults in a ring other than rings 0 and 1. Hence there are at least 3 rings containing faults and consequently the number of faults is at most 2 in each ring. In this case *Ada-RingTest* can be applied to complete testing of all horizontal rings. If there are 4 rings containing faults, there is a unique fault in each ring and it can be discovered using m tests per ring, for a total of km tests. If there are 3 rings containing faults then then *Ada-RingTest* uses at most $m+1$ tests per faulty ring, for a total of $km+3$ tests.

Case 4: Rings 0, 1 and 2 contain faulty nodes. The argument is the same as in Case 3.2. ■

The remaining cases are those of the 3×3 -torus, 3×4 -torus and 4×4 -torus. The last two of them are similar to the above, with two exceptions. On the one hand, 2 faults are not always possible to diagnose in a ring of length 4, without using testers from outside. On the other hand, a healthy ring of length 4 may consist exclusively of faulty nodes. Nevertheless, diagnosis can always be done using at most 15 tests for the 3×4 -torus and at most 19 tests for the 4×4 -torus. These worst-case numbers of tests are optimal in view of the lower bound from [2]. We omit the details of testing and diagnosis in these cases.

Corollary 4.4 *A $k \times m$ -torus, for $k \geq 3$ and $m \geq 4$, can be correctly diagnosed using at most $km + 3$ tests in the worst case, provided that the number of faults is at most 4. This number of tests is optimal.*

The only case we have not considered is that of the 3×3 -torus. We do not know if all configurations of at most 4 faults in this torus can be diagnosed using at most 12 tests, which corresponds to the lower bound from [2] in this case. We leave this as an open problem.

The following proposition shows the difference between the number of tests needed to perform diagnosis of a torus in the nonadaptive and in the adaptive way.

Proposition 4.5 *The minimum number of tests sufficient to perform one-step diagnosis of at most 3 faults in a $2 \times m$ -torus is $6m$, and to perform one-step diagnosis of at most 4 faults in a $k \times m$ -torus, for $k, m \geq 3$ is $4km$.*

Proof: Straightforward. ■

5 Conclusion

We have shown optimal adaptive diagnosis algorithms for systems represented by trees, rings and tori. For trees, the maximum number of faults was assumed to be 1, for rings 2, for $2 \times m$ -tori 3, and for other tori 4. These restrictions on the number of faults are necessary due to low minimum degree of the above graphs (1 for trees and 2 for rings, 3 or 4 for tori). We showed that in all cases the optimum worst-case number of tests in adaptive diagnosis is smaller than that in one-step diagnosis.

Our result for trees implies that adaptive 1-diagnosis using n tests in the worst case is possible for any system represented by a connected graph. On the other hand, the result for rings implies adaptive 2-diagnosis using $n + 1$ tests for any system represented by a Hamiltonian graph. The efficiency gain due to adaptive testing remains valid for 2-diagnosis of any Hamiltonian graph. Any such graph (in fact, any 2-diagnosable graph) requires at least $2n$ tests in one-step diagnosis. In contrast, one-step diagnosis can be done as efficiently as adaptive diagnosis (i.e., using n tests) for some graphs (e.g., for rings).

References

[1] M. Barborak, M. Malek, A. Dahbura, The consensus problem in fault-tolerant Computing, *ACM Computing Surveys*, Vol. 25, pp. 171 - 220, 1993.

[2] P. M. Blecher, On a logical problem, *Discrete Mathematics 43*, pp. 107 - 110, 1983.

[3] D.M. Blough, G.F. Sullivan and G.M. Masson, Efficient diagnosis of multiprocessor systems under probabilistic models, IEEE Transactions on Computers 41 (1992), 1126-1136.

[4] S. L. Hakimi and A. T. Amin, Characterization of connection assignment of diagnosable systems, *IEEE Tans. on Computers 23*, pp. 86 - 88, 1974.

[5] S. L. Hakimi and K. Nakajima, On adaptive system diagnosis, *IEEE Trans. on Computers 33*, pp. 234 - 240, 1984.

[6] E. Kranakis and A. Pelc, Better adaptive diagnosis of hypercubes, to appear.

[7] G. Masson, D. Blough, and G. Sullivan, "System Diagnosis", in *Fault-Tolerant Computer System Design*, D. K. Pradhan, ed., Prentice-Hall, 1996.

[8] K. Nakajima, A new approach to system diagnosis, *Proc. 19th Annu. Allerton Conf. Commun., Contr. and Comput.*, pp. 697 - 706, Sept. 1981.

[9] A. Pelc, Undirected graph models for system-level fault diagnosis, *IEEE Trans. on Computers 40*, pp. 1271 - 1276, 1991.

[10] A. Pelc and E. Upfal, Reliable fault diagnosis with few tests, *Combinatorics, Probability & Computing* 7 (1998), 323-333.

[11] F. P. Preparata, G. Metze, and R. T. Chien, On the connection assignment problem of Diagnosable Systems, *IEEE Trans. Electronic Computers*, no. 12, pp. 848 - 854, Dec. 1967.

[12] C. -C. Wu, Partial solution to problem 81-6, *Journal of Algorithms*, Vol. 3, pp. 379 - 380, 1982.