System Diagnosis

- Objective
  - Designing systems that are capable of self-diagnoses of multiple faults
- Motivation
  - Multiprocessor systems employ increasing numbers of processors. Some of these processors will fail.
  - Applications include safety critical systems.
  - Inaccessible systems, e.g. remote, under water or ground, space.
  - "It is always good to know who your enemies are".

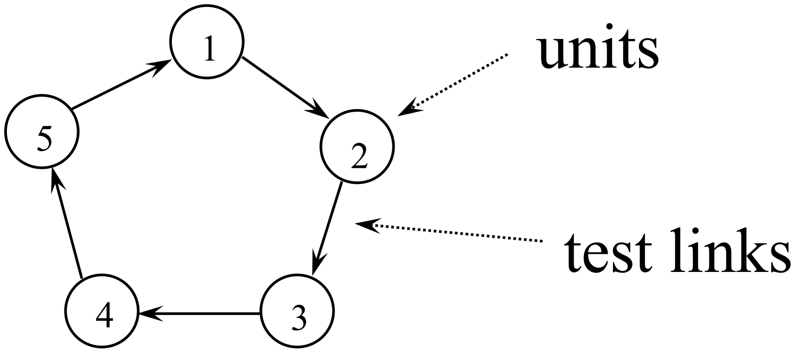
System Diagnosis

- Assumptions
  - System is partitioned into *units*
    - units need not be identical
    - units are powerful enough to test and judge other units pass/fail.
  - Tests are adequate to detect all faults
    - perfect coverage. (This is very restrictive since it also implies faults to be permanent).
  - There exists a reliable method for collecting and evaluating all test results
    - e.g. reliable broadcast
  - These assumptions are often termed *PMC Model*, after early work by Preparata, Metze and Chien (1967)

    "On the Connection Assignment Problem of Diagnosable Systems", IEEE Trans. on Electronic Computers, Vol. EC16, No. 6, Dec. 1967.

System Diagnosis

- System Graph



- Definitions
  - Test graph $G = (U, E)$
  - U : the set of units
  - E : the set of testing links (edges)
  - a_{ij}: the outcome of test (U_i, U_j)

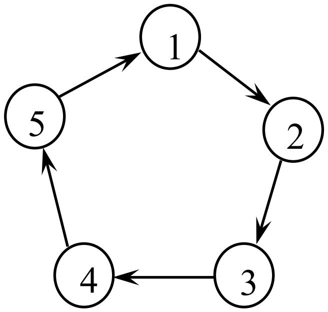
if U_i is non-faulty then

    if U_j is non-faulty => $a_{ij} = 0$

    if U_j is faulty => $a_{ij} = 1$

else a_{ij} is unreliable

System Diagnosis



- Example: single fault, U_1 faulty
  - Then $a_{51} = 1$ and $a_{12} = X$ (0 or 1)
  - Syndrome S = set of all outcomes
    - order all a_{ij}

$$-> a_{12} \quad -> \quad a_{23} \quad -> \quad a_{34} \quad -> \quad a_{45} \quad -> \quad a_{51} \quad ->\wedge$$

$$-> X \quad -> \quad 0 \quad -> \quad 0 \quad -> \quad 0 \quad -> \quad 1 \quad ->\wedge$$

  - 2 cases
    - Single 1 in a_{51} => U_1 is faulty
      - note if U_5 was faulty, then $a_{45} = 1$
    - Pair of adjacent 1's
      - the "upstream" 1 is correct
      - $a_{45} = 1$

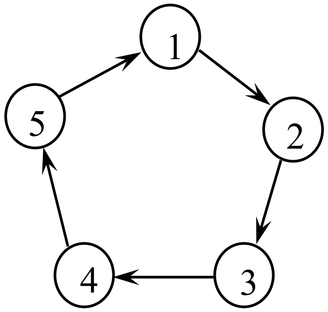
System Diagnosis

- Definition: t-fault-diagnosable
  - Every set of up-to t faulty units can be correctly diagnosed (eventually).
    - Previous example is 1-fault-diagnosable
    - not 2-fault diagnosable
      - e.g. assume U1 and U2 faulty and $a_{12} = 0$
      - same syndrome as 1-fault-diagnosable example
- Definition: one-step t-fault-diagnosable
  - For every set of up-to *t* faulty units there exists a unique syndrome which correctly identifies <u>all</u> faulty units.

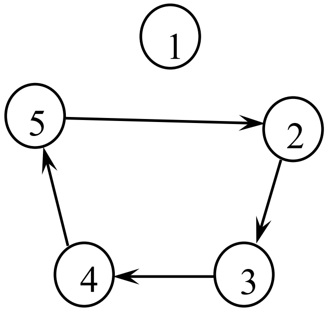
System Diagnosis

- Definition: Sequential t-fault-diagnosable
  - For every set of up-to t faulty units there exists a unique syndrome which correctly identifies at least one faulty unit.
  - (Can be applied recursively)

System Diagnosis

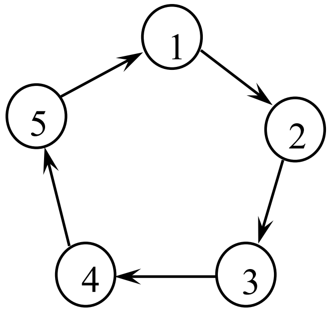


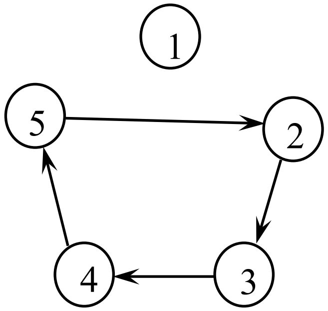
- Example: dual fault
  - Assume U_1 and U_2 are both faulty
    - $(a_{12}\ , a_{23}\ , a_{34}\ ,\ \ a_{45}\ , a_{51}\ )$
    - (X , X , 0 , 0 , 1)
  - Could mimic single fault at U_1
    - i.e. (0 , 0 , 0 , 0 , 1)
  - But, pattern 001 always points to a faulty unit
    - thus remove U_1 and reconfigure
    - $(a_{23}\ , a_{34}\ ,\ \ a_{45}\ , a_{52}\ )$
    - (X , 0 , 0 , 1)
    - still have 001 pattern => U_2 diagnosed to be faulty



System Diagnosis

- Example: dual fault
  - Assume U_1 and U_3 are both faulty
    - $(a_{12}, a_{23}, a_{34}, a_{45}, a_{51})$
    - (X , 1 , X , 0 , 1)
  - Now possible pattern
    - (1 , 1 , 1 , 0 , 1)
    - still points to one faulty unit => U_1
    - after reconfiguration
      - $(a_{23}, a_{34}, a_{45}, a_{52})$
      - (1 , X , 0 , 0)
    - points to one faulty unit => U_3





System Diagnosis

- Necessary and Sufficient Conditions
  - » t = # of faults
  - » n = # of units
  - » N = # of testing links
  - – One-step t-fault-diagnosable system

    $$n \geq 2t + 1$$

    - » each unit is tested by **<u>more</u>** than t-1 <u>other</u> units (or: at least t)
    - » this implies $N \geq n \cdot t$
    - » optimal: replace $\geq$ with $=$
  - – Sequentially t-fault-diagnosable system
    - » $n \geq 2t + 1 \longleftarrow$ necessary
    - » $N \geq n + 2t - 2 \longleftarrow$ and sufficient

System Diagnosis

- Single Loop System (Ring)
  - Let $t = 2\underset{\substack{\uparrow \\ \text{integer}}}{m} + \underset{\substack{\nwarrow \\ \text{0 or 1 (even or odd)}}}{\lambda}$
  - Loop is sequentially t-fault-diagnosable if

    $$n \geq 1 + (m+1)^2 + \lambda(m+1)$$

    - proof given in paper Pre67
  - e.g.
    - $t = 1 \Rightarrow \quad m = 0, \quad \lambda = 1, \qquad n = 1 + 1 + 1 = 3$
    - $t = 2 \Rightarrow \quad m = 1, \quad \lambda = 0, \qquad n = 1 + 2^2 + 0 = 5$
    - $t = 3 \Rightarrow \quad m = 1, \quad \lambda = 1, \qquad n = 1 + 2^2 + 2 = 7$

System Diagnosis

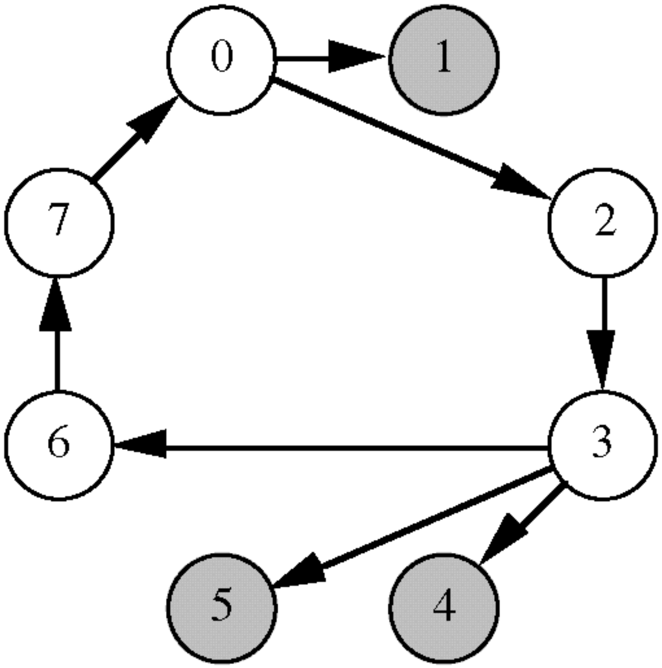
- Inefficiency of PMC
  - PMC requires for t-diagnosability, that each node must be tested by at least *t* other nodes.
  - Problem: many diagnosis!
  - Alternative: adaptive models
    - the term adaptive stems from allowing the choice of which test(s) to perform depend on the results of previous tests.

System Diagnosis

- Adaptive Distributed System Diagnosis
    - » "Implementation of On-Line Distributed System-Level Diagnosis Theory" by Bianchini and Buskens, Trans. on Computers, May 1992
  - Uses array TESTED_UP$_x$ at each node n_x
  - Meaning of TESTED_UP$_x$[i]
    - » TESTED_UP$_x$[i] = j implies that node n_x has received information from fault-free node n_i, that n_i found n_j to be fault-free.
  - Idea:
    - » each node finds first node that is fault-free
      - ■ n_i checks n_j $j > i \bmod N$, where N is the number of nodes.
    - » get other TESTED_UP$_i$ values from TESTED_UP$_j$
    - » implies that node n_x has received information from fault-free node n_j, that n_i found n_j to be fault-free.

System Diagnosis

Example: assume nodes 1, 4 and 5 are faulty



$\text{TESTED\_UP}_2[0] = 2$

$\text{TESTED\_UP}_2[1] = x$

$\text{TESTED\_UP}_2[2] = 3$

$\text{TESTED\_UP}_2[3] = 6$

$\text{TESTED\_UP}_2[4] = x$

$\text{TESTED\_UP}_2[5] = x$

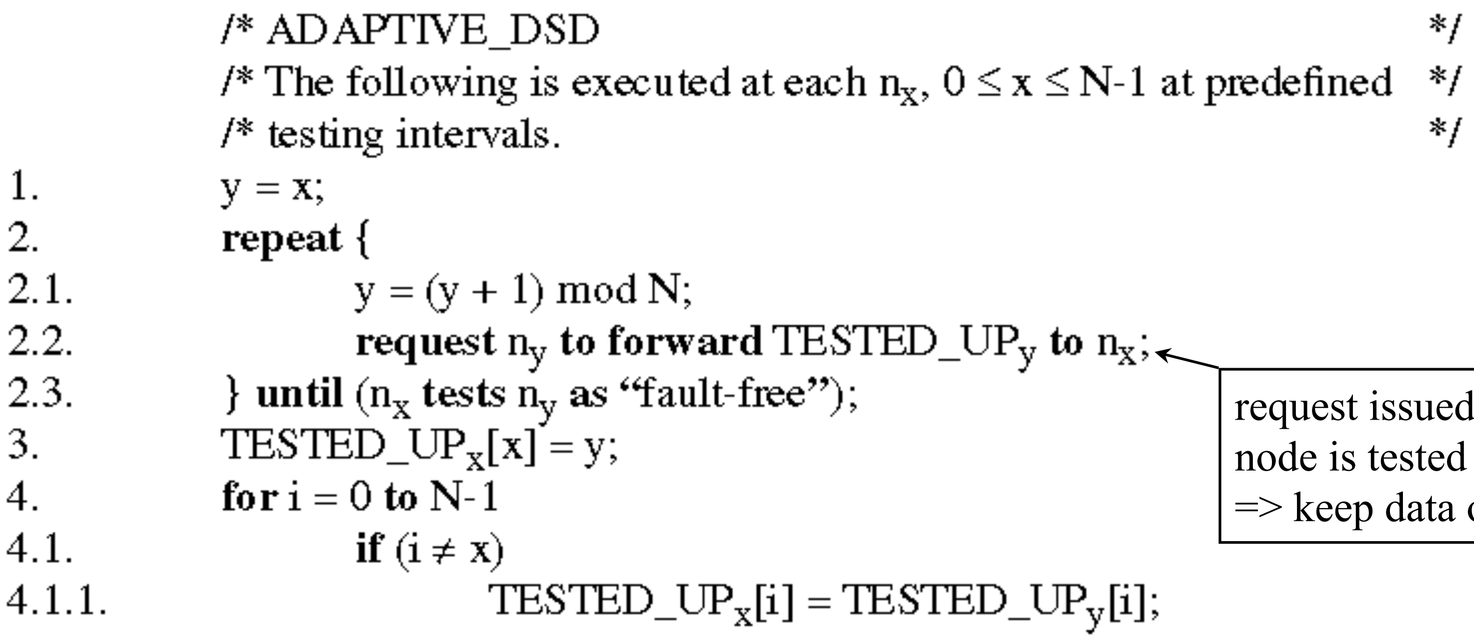
$\text{TESTED\_UP}_2[6] = 7$

$\text{TESTED\_UP}_2[7] = 0$

Bia92, fig 4 and 6

System Diagnosis

- Adaptive Distr.Sys.Diag. Algorithm (Bia92 fig 5)



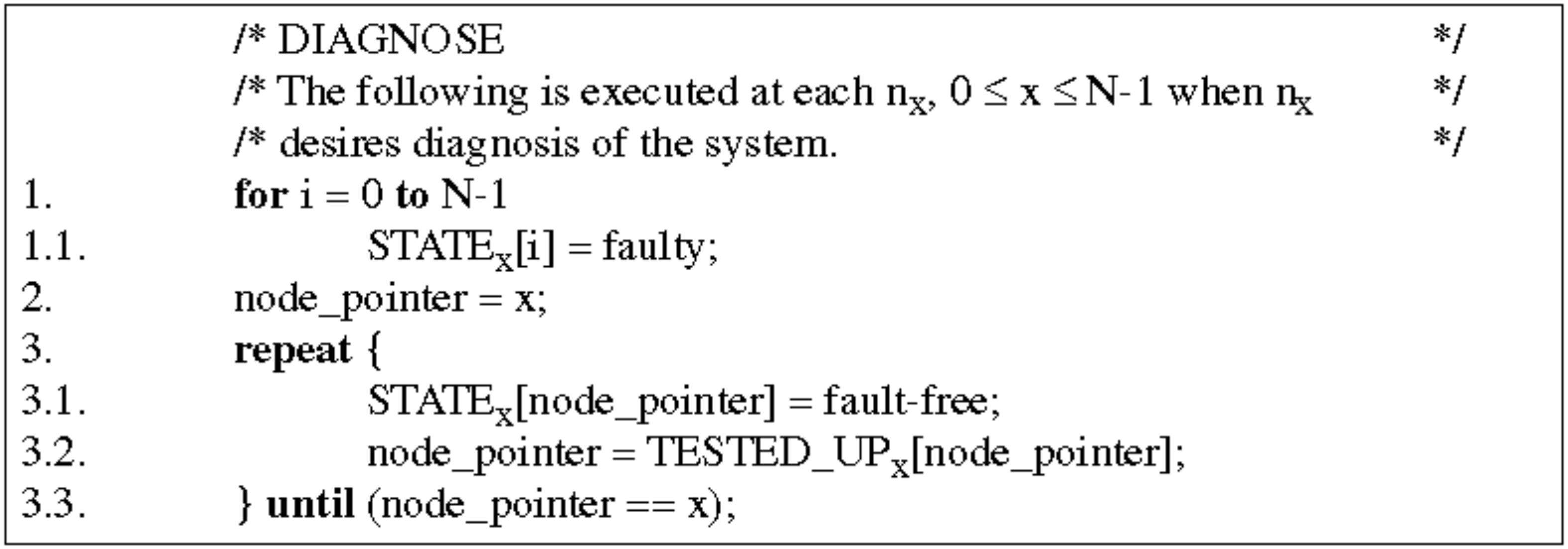
```
         /* ADAPTIVE_DSD                                                         */
         /* The following is executed at each $n_x$, $0 \le x \le N-1$ at predefined   */
         /* testing intervals.                                                   */
1.       $y = x$;
2.       repeat {
2.1.         $y = (y + 1) \bmod N$;
2.2.         request $n_y$ to forward TESTED_UP$_y$ to $n_x$;
2.3.     } until ($n_x$ tests $n_y$ as "fault-free");
3.       TESTED_UP$_x[x] = y$;
4.       for $i = 0$ to $N-1$
4.1.         if $(i \ne x)$
4.1.1.           TESTED_UP$_x[i]$ = TESTED_UP$_y[i]$;
```

request issued before
node is tested fault-free
=> keep data only if n_y ok

(Bia92 fig 5)

System Diagnosis

- Diagnosis
  - accomplished at any node n_x by following the fault-free paths from n_x to other fault-free nodes.



```
/* DIAGNOSE                                                        */
/* The following is executed at each $n_x$, $0 \leq x \leq N-1$ when $n_x$  */
/* desires diagnosis of the system.                                */
for i = 0 to N-1
        STATE_x[i] = faulty;
node_pointer = x;
repeat {
        STATE_x[node_pointer] = fault-free;
        node_pointer = TESTED_UP_x[node_pointer];
} until (node_pointer == x);
```

Bia92 fig. 7