

FAULT TOLERANT SYSTEMS

<http://www.ecs.umass.edu/ece/koren/FaultTolerantSystems>

Part 12 - Networks - 2

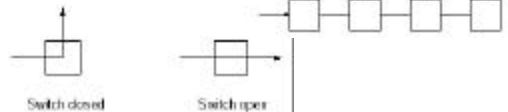
Chapter 4: Network Fault Tolerance

Part.12 .1

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Crossbar Networks

- ◆ In a multistage network
 - processors competing over links
 - bandwidth limited
- ◆ **Crossbar**
 - higher bandwidth
 - a switchbox for each input/output pair
 - * $N \times M$ crossbar - N inputs, M outputs, NM switches
- ◆ The (i,j) switchbox connects row i input to column j output and is capable of
 - * propagating a message along row from left link to right link
 - * Propagating a message along column from bottom to top link
 - * turning a message from left link to top link
- ◆ A link carries one message; a switch can process up to two messages at the same time
- ◆ **Crossbar is not fault-tolerant** - failure of any switchbox will disconnect certain pairs



Part.12 .2

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Routing in Crossbar

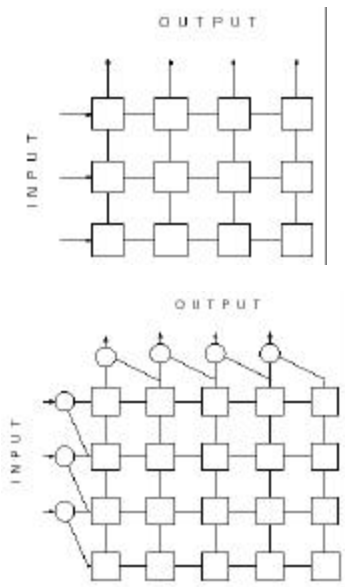
- ◆ **Example:** Sending a message from input 3 to output 5:
- ◆ Source \rightarrow switchbox (3,1) \rightarrow (3,2) \rightarrow ... \rightarrow (3,5) \rightarrow (2,5) \rightarrow (1,5) \rightarrow destination
- ◆ High bandwidth - any input-output combination can be realized if no two inputs are competing for access to same output line
- ◆ High bandwidth especially desirable when both inputs and outputs are connected to high-speed processors
- ◆ Cost of high performance - $N \times M$ crossbar has NM switchboxes ; $N \times N$ multistage network has only $N^{1/2} \cdot \log_2 N$ switchboxes

Part.12 .3

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Redundant Crossbar

- ◆ Adding redundancy to make the crossbar fault-tolerant:
- ◆ A row and a column of switches are added
 - * Input and output connections are augmented - each input can be sent to either of two rows and each output can be received on either of two columns
- ◆ If a switch becomes faulty - row and column to which it belongs are replaced by the spare row and column



Part.12 .4

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Original Crossbar Connectability

- ◆ q_l - probability that a link is faulty

$$p_l = 1 - q_l$$

- ◆ Probability of switchbox failures included in link failure probabilities
- ◆ For input i to be connectable to output j , we have to go through $i+j$ links

- ◆ Therefore -

$$Q = \sum_{i=1}^N \sum_{j=1}^M p_l^{i+j} = p_l^2 \frac{1 - p_l^N}{1 - p_l} \frac{1 - p_l^M}{1 - p_l}$$

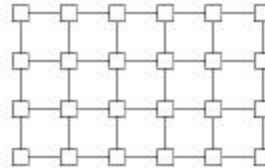
- ◆ **Exercise:** Calculate Q for the fault-tolerant crossbar and the bandwidth BW for both designs

Part.12 .5

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Mesh Networks

- ◆ **2-dimensional NxM rectangular mesh network**



- * All nodes computing
- * No separate switchboxes
- ◆ **Mesh property** - most NM computing nodes (except boundary ones) have **4** incident links and **4** neighbors
- ◆ Sending a message - a path from source to destination identified and message forwarded along path
- ◆ **Conventional mesh** - even one fault will disrupt the **mesh property** (necessary for some algorithms)
- ◆ **Conventional mesh reliability:** $R_{\text{mesh}}(t) = [R(t)]^{NM}$
 - * $R(t)$ - reliability of one node
- ◆ To increase reliability - introduce redundancy

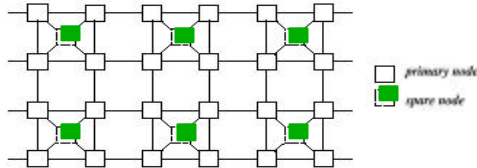
Part.12 .6

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Interstitial Mesh

- ◆ Conventional 2-dimensional rectangular mesh network - unable to tolerate any faults in a node

- ◆ (1,4) Interstitial Redundancy



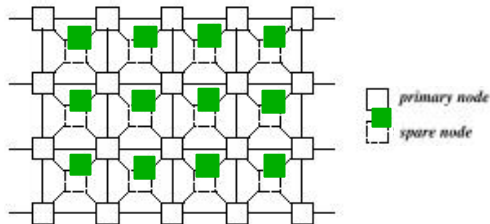
- ◆ A spare node can be switched in to replace any neighbor that failed
- ◆ Each primary node has a single spare node while each spare node is a spare for four primary nodes
- ◆ Redundancy overhead is 25%
- ◆ Main advantage - physical proximity of spare node to the primary node which it replaces - reducing delay penalty due to use of a spare

Part.12 .7

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Different Interstitial Redundancy

- ◆ (4,4) Interstitial Redundancy



- ◆ Primary node has four spare nodes
- ◆ Each spare node is a spare for four primary nodes
- ◆ Higher level of fault tolerance - higher redundancy overhead of almost 100%

Part.12 .8

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Reliability of Mesh with (1,4) Interstitial Redundancy

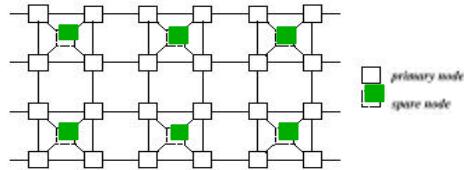
- ◆ Mesh is of size $N \times M$ with N, M even numbers
- ◆ Cluster - four primary nodes with one spare node
- ◆ Mesh has $NM/4$ clusters
- ◆ $R(t)$ - reliability of primary or spare node
- ◆ Reliability of cluster

$$R_{cluster} = R^5(t) + 5R^4(t)[1 - R(t)]$$

- ◆ Reliability of mesh

$$R_{mesh}(t) = [R_{cluster}(t)]^{NM/4}$$

- ◆ No simple algorithm to calculate reliability of the (4,4) interstitial scheme



Part.12 .9

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Another Mesh Dependability Measure

- ◆ An application that is about to run on the mesh requires an $n \times m$ submesh ($n < N$ and $m < M$)
- ◆ Dependability measure - probability of being able to allocate an $n \times m$ fault-free submesh
 - * Computation of this probability is very difficult
- ◆ **Approximation:** only non-overlapping predetermined submeshes can be allocated

- ◆ Number of possible allocations now limited to

$$k = \lfloor N/n \rfloor \times \lfloor M/m \rfloor$$

Prob{a fault-free $n \times m$ submesh can be allocated}

$$= 1 - [1 - (R(t))^{nm}]^k$$

- * $R(t)$ - reliability of a node
- * If nodes can be repaired - availability more appropriate

Part.12 .10

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Hypercube Networks

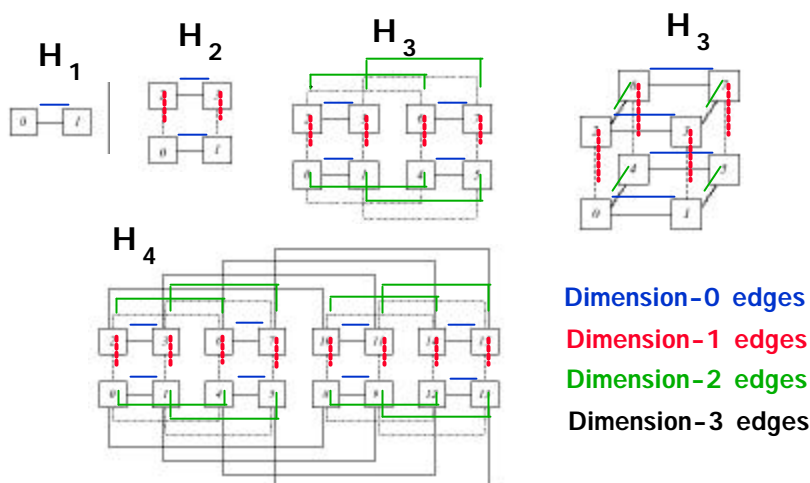
- ◆ H_n - An n -dimensional hypercube network - 2^n nodes
- ◆ A 0-dimensional hypercube H_0 - a single node
- ◆ H_n constructed by connecting the corresponding nodes of two H_{n-1} networks
- ◆ The edges added to connect corresponding nodes are called **dimension-(n-1) edges**
- ◆ Each node in an n -dimensional hypercube has n edges incident upon it



Part.12 .11

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Hypercube - Examples



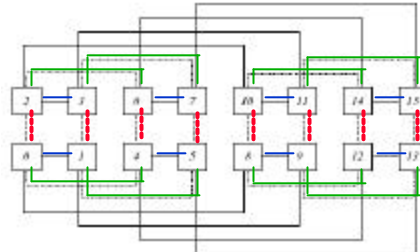
Part.12 .12

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Routing in Hypercubes

- ◆ Specific numbering of nodes to simplify routing
- ◆ Number expressed in binary - if nodes i and j are connected by a **dimension- k** edge, the names of i and j differ in only the k -th bit position
- ◆ **Example** - nodes **0000** and **0010** differ in only the 2^1 bit position - connected by a **dimension-1** edge
- ◆ **Example** - a packet needs to travel from node **14=1110₂** to node **2=0010₂** in an **H₄** network
- ◆ **Possible routings** -
 - * **1110** ® **0110** (dimension 3)
 - ® **0010** (dimension 2)
 - * **1110** ® **1010** (dimension 2)
 - ® **0010** (dimension 3)

Dimension-0 edges
 Dimension-1 edges
 Dimension-2 edges
 Dimension-3 edges



Part.12 .13

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Routing in Hypercubes - General

- ◆ **In general** - distance between source and destination is the number of different bits in binary addresses
- ◆ Going from X to Y can be accomplished by traveling once along each dimension in which they differ

$$X = x_{n-1} \dots x_0$$

$$Y = y_{n-1} \dots y_0$$

Define $z_i = x_i \oplus y_i$

where \oplus is the **exclusive-or** operator

- ◆ Packet must traverse an edge in every dimension i for which $z_i = 1$

Part.12 .14

Copyright 2007 Koren & Krishna, Morgan-Kaufman

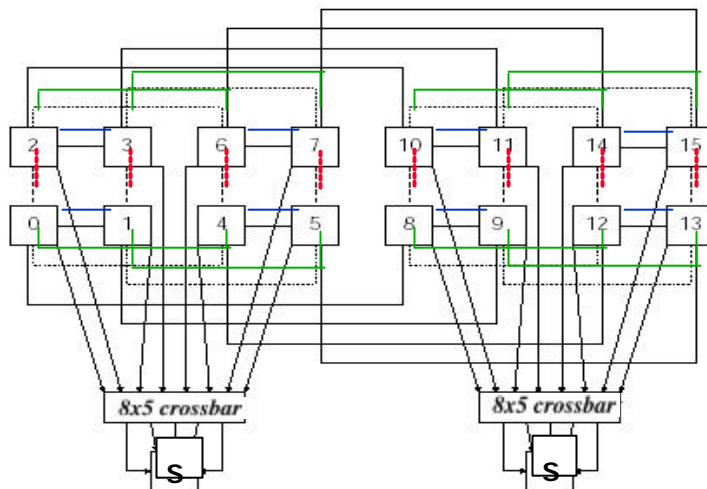
Adding Fault Tolerance to Hypercubes

- ◆ H_n (for $n \geq 2$) can tolerate link failures
 - multiple paths from any source to any destination
- ◆ Node failures can disrupt the operation
- ◆ Adding fault tolerance:
 - * Adding one or more spare nodes
 - * increasing number of communication ports of each original node from n to $n+1$
 - * connecting the extra ports through additional links to spare nodes
- ◆ Example - two spare nodes - each a spare for 2^{n-1} nodes of an H_{n-1} sub-cube
- ◆ Spare nodes may require 2^{n-1} ports
- ◆ Using crossbar switches with outputs connected to spare node reduces number of ports of spare node to $n+1$ - same as all other nodes

Part.12 .15

Copyright 2007 Koren & Krishna, Morgan-Kaufman

An H_4 Hypercube with Two Spare Nodes

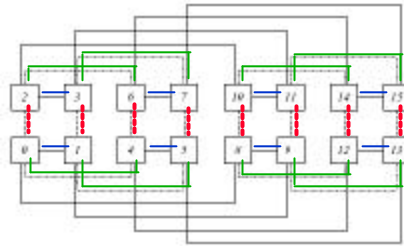


Each node is of degree 5

Part.12 .16

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Different Method of Fault-Tolerance



- ◆ Duplicating the processor in a few selected nodes
- ◆ Each additional processor - spare also for any of the processors in the neighboring nodes
- ◆ **Example** - nodes 0, 7, 8, 15 in H_4 - modified to duplex nodes
- ◆ Every node now has a spare at a distance no larger than 1
- ◆ Replacing a faulty processor by a spare results in an additional communication delay

Part.12 .17

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Reliability of A Hypercube

- ◆ **Assumption:** nodes and links all fail independently
- ◆ Reliability of H_n is the product of
 - * Reliability of 2^n nodes, and
 - * Probability that every node can communicate with every other node
- ◆ Exact evaluation of this probability difficult - every source-destination pair connected by multiple paths
- ◆ Instead - we obtain a good **lower bound** on the reliability
- ◆ Exploiting recursive nature of hypercube - we add probabilities of three mutually exclusive cases for which the network is connected
- ◆ This is a lower bound - there may be other cases where H_n is connected

Part.12 .18

Copyright 2007 Koren & Krishna, Morgan-Kaufman

Three Mutually Exclusive Cases

- ◆ Decompose H_n into two H_{n-1} hypercubes, A and B , and the **dimension-(n-1)** links connecting them
- ◆ **Case 1:** Both A and B are operational and at least one **dimension-(n-1)** link is functional
- ◆ **Case 2:** One of A, B is operational and the other is not, and all **dimension-(n-1)** links are functional
- ◆ **Case 3:** Only one of A, B is operational, exactly one **dimension-(n-1)** link is faulty and is connected in the nonoperational H_{n-1} to a node that has at least one functional link to another node
- ◆ **Exercise:** show that each of these cases results in a connected H_n and that they are mutually exclusive

Probabilities of the Three Cases

- ◆ **Notations:** q_c - probability of a node failure
 q_l - probability of a link failure
 $NR(H_n, q_l, q_c)$ - reliability of hypercube H_n
 * t omitted for simplicity

- ◆ **Assumption:** Nodes are perfectly reliable ($q_c=0$)

* This assumption will be modified later

- ◆ **Case 1:** Both A and B are operational and at least one **dimension-(n-1)** link is functional

$$\text{Prob}\{\text{Case 1}\} = [NR(H_{n-1}, q_l, 0)]^2 (1 - q_l^{2^{n-1}})$$

- ◆ **Case 2:** One of A, B is operational and the other is not and all **dimension-(n-1)** links are functional

$$\text{Prob}\{\text{Case 2}\} = 2NR(H_{n-1}, q_l, 0)[1 - NR(H_{n-1}, q_l, 0)](1 - q_l)^{2^{n-1}}$$

- ◆ **Case 3:** Only one of A, B is operational, exactly one **dimension-(n-1)** link is faulty and is connected in the nonoperational H_{n-1} to a node that has at least one functional link to another node

$$\begin{aligned} \text{Prob}\{\text{Case 3}\} &= 2NR(H_{n-1}, q_l, 0)[1 - NR(H_{n-1}, q_l, 0)] \\ &\quad \times 2^{n-1} q_l (1 - q_l)^{2^{n-1} - 1} (1 - q_l^{n-1}) \end{aligned}$$

Reliability of H_n

$$NR(H_n, q_e, 0) = \text{Prob}\{\text{Case 1}\} + \text{Prob}\{\text{Case 2}\} + \text{Prob}\{\text{Case 3}\}$$

◆ **Initial case :**

- * either hypercube of **dimension 1**: two nodes and one link

$$NR(H_1, q_e, 0) = 1 - q_e$$

- * or, hypercube of **dimension 2**

$$NR(H_2, q_e, 0) = (1 - q_e)^4 + 4q_e(1 - q_e)^3$$

- ◆ The results will be different in both cases

- ◆ If the nodes are not perfect ($q_c \neq 0$)

$$NR(H_n, q_e, q_c) = (1 - q_c)^{2^n} NR(H_n, q_e, 0)$$