Fault Tolerant Systems

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

Part 12 - Networks - 2

Chapter 4: Network Fault Tolerance
(Modified for ECE753 – Saluja)

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

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 1st bit position - connected by a dimension-1 edge
- Example - a packet needs to travel from node $14=1110_2$ to node $2=0010_2$ in an $H_4$ network

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 nodes of an $H_{n-1}$ sub-cube
- Spare nodes may require $n+1$ 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
An H4 Hypercube with Two Spare Nodes

Each node is of degree 5

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 H4 - 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

Reliability of A Hypercube

♦ Assumption: nodes and links all fail independently
♦ Reliability of Hn 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 Hn is connected

Three Mutually Exclusive Cases

♦ Decompose Hn into two Hn-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 Hn-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 Hn and that they are mutually exclusive

Reliability of Hn

NR(Hn, q_s, q_c) = Prob{Case 1} + Prob{Case 2} + Prob{Case 3}
♦ Initial case:
  ♦ either hypercube of dimension 1: two nodes and one link
    NR(H_1, q_s, q_c) = 1 - q_c
  ♦ or, hypercube of dimension 2
    NR(H_2, q_s, q_c) = (1 - q_c)^4 + 3q_c(1 - q_c)^3
♦ The results will be different in both cases
♦ If the nodes are not perfect (q_c ≠ 0)
  NR(H_n, q_s, q_c) = (1 - q_c)^2NR(H_n, q_s, 0)