Related Works:

Author(s)	**Year**	**Overview**	**Analysis**	**Observation**
P. Ramanathan and K. G. Shin	1988	A simple algorithm for broadcasting in a hypercube multicomputer containing faulty nodes/links was proposed. The algorithm delivers multiple copies of the broadcast message through disjoint paths to all the modes in the system.	Its salient feature is that the delivery of the multiple copies is transparent to the processes receiving the message and does not require the processes to know the identity of the faulty processors. The processes on non-faulty nodes that receive the message identify the original message from the multiple copies using some scheme appropriate for the fault model used.	The algorithm completes in n +1 steps if each node can simultaneously use all of its outgoing links. If each node cannot use more than one outgoing link at a time, then the algorithm requires 2n steps.
Y. Lan	1992	Multicast algorithms for faulty hypercube multicomputers are discussed. Two types of algorithms were proposed.	Type I algorithms have the following features: they are distributed, in the sense that the same algorithm is implemented in all involved nodes and based on local information only; they can always find shortest paths from the source to all destinations whenever such exist. Type II algorithms deal with both link faults and node faults. They can deliver messages to all reachable destinations if the total number of faults is less than the dimension of the hypercube.	Type I algorithms deal with nodes faults only, and they cannot deliver messages to those destinations that can be reached through some longer paths. These algorithms are not easy to implement in hardware.

Jehoshua Bruck	1994	This paper presented n spanning trees of the n-cube with the following properties: (i) All the tree in H_n are rooted at the same node (e.g., the all-0 node). (ii) Every tree in H_n is of depth n, which is the optimal depth. (iii) Every edge in the n-cube is shared by at most two trees in H_n.	One of the important application of spanning tree construction is to perform broadcasting in a hypercube with at most $\lceil n/2 \rceil$ -1 faulty links.	The main problem of this approach was that it can not deal with $\lceil n/2 \rceil$ -1 faulty links.
Y. R. Lan	1994	Fault-tolerant message routing mechanism is a key to the performance of reliable multicomputers. Multicast refers to the delivery of the same message from a source node to an arbitrary number of destination nodes.	This paper presented two types of partially adaptive fault tolerant multicast algorithms. The Type A algorithm can deliver messages to all destinations through shortest paths if each fault-free node has at most one faulty neighbor. The Type B algorithm can deliver messages to all destinations if the total number of faulty links and faulty nodes is less than the dimension of the hypercube.	The proposed algorithms have the following important features: they are distributed, they only require local information to determine the paths, and they need very little additional message overhead.
Q.-P. Gu and S. Peng	1996	This paper presented an algorithm which finds path from source to destination in the presence of n-1 faulty nodes and non-faulty nodes in H_n.	Using this algorithm as a subroutine, they presented another algorithm which, given at most 2n-3 faulty nodes such that the faulty nodes can be covered by n-1 subgraphs of diameter 1.	The proposed algorithm finds a fault-free path from source to destination of length at most d(s, t)+4 in O(n) time.

Seungjin Park and Bella Bose	1997	This paper presented a fault-tolerant all-to-all broadcasting algorithms in H_n with up to $\lfloor n/2 \rfloor$ faulty links/nodes. The proposed algorithm assume a multiport I/O model, meaning each node can send and receive messages from all its adjacent simultaneously.	The proposed algorithms have several desirable features such as (i) each node sends only one copy of the broadcast message, which reduces traffic in the network. (ii) they utilize an algorithm developed for the non-faulty system. (iii) they achieve near optimal performance.	They presented two new fault-tolerant all-to-all broadcasting (or multimode broadcasting) algorithms that can tolerate up to $\lfloor n/2 \rfloor$ links or $\lfloor n/2 \rfloor$ node faults in a H_n.
Jie Wu	1997	They proposed a unicasting algorithm for faulty hypercubes (including disconnected hypercubes) using the safety level concept. A faulty hypercube is a hypercube with faulty nodes and unicasting is a one-to-one communication between two nodes in the hypercube. Each node is associated with a safety level which is an approximated measure of the number and distribution of faulty nodes in the neighborhood. The safety level of each node in an n-dimensional hypercube can be easily calculated through n-1 rounds of information exchange among neighboring nodes.	Optimal unicasting between two nodes is guaranteed if the safety level of the source node is no less than the Hamming distance between the source and the destination. The proposed unicasting algorithm can also be used in disconnected hypercubes, where nodes in a hypercube are disjointed into two or more parts. The feasibility of an optimal or suboptimal unicasting can be easily determined at the source node by comparing its safety level, its neighbors' safety levels, and the Hamming distance between the source and the destination.	The proposed scheme is the first attempt to address the unicasting problem in disconnected hypercubes. The safety level concept is also extended to be used in hypercubes with both faulty nodes and links and in generalized hypercubes
Ge-Ming Chiu and Kai-Shung	1998	This paper presented a fault-tolerant multicast scheme for hypercube multicomputers. The method is	The routing capability information is used to guide derouting in an efficient manner when such needs	Two multicast algorithms are presented in the paper. These algorithms multicast messages

Chen		based on the routing capability information that is stored in each node. This information is able to capture the fault status more precisely.	arise. The amount of traffic incurred is addressed in the paper. The hardware design for the algorithms is also discussed.	in an attempt to minimize derouting so that time optimality can be achieved.
Qian-Ping Gu and S. Peng	1999	Unicast in computer/communication networks is a one-to-one communication between a source node s and a destination node t. They proposed three algorithms which find a non-faulty routing path between s and t for unicast in the hypercube with a large number of faulty nodes.	First algorithm finds a non-faulty path of length at most d(s,t)+4 in O(n) time for unicast between 1-safe s and t in the H_n with $\lvert F \rvert \leqslant 2n-3$, where d(s,t) is the distance between s and t. The second algorithm finds a non-faulty path of length at most d(s,t)+6 in O(n) time for unicast in the 2-safe H_n with $\lvert F \rvert \leqslant 4n-9$. The third algorithm finds a non-faulty path of length at most $d(s,t)+O(k^2)$ in time O(|F|+n) for unicast in the k-safe H_n with $\lvert F \rvert \leqslant 2^k(n-k)-1$ $(0 \leqslant k \leqslant n/2)$.	The time complexities of the algorithms are optimal. They showed that in the worst case, the length of the non-faulty path between s and t in a k-safe H_n with $\lvert F \rvert \leqslant 2^k(n-k)-1$ is at least d(s,t)+2(k+1) for $0 \leqslant k \leqslant n/2$. This implies that the path lengths found by the algorithms for unicast in the 1-safe and 2-safe hypercubes are optimal.
Dong Xiang	2001	This paper studies fault-tolerant routing for injured hypercube using local safety information.	A routing scheme based on local safety information is proposed and the extra cost to obtain local safety information is comparable to the one based on global safety information.	The proposed algorithm guarantees to find a minimum feasible path if the spanning subcube is contained in a maximal safe subcube and the source is locally safe in the maximal safe subcube.

Shyue-Ming Tang, Yue-Li Wang, and Yung-Ho Leu	2004	Two spanning trees rooted at some vertex R in a graph G are said to be independent if for each vertex v of G, they have unique trees.	A set of independent spanning trees is optimal if the average path length of the trees is the minimum.	Any k-dimensional hypercube has k independent spanning trees rooted at an arbitrary vertex. In this paper, an O(kn) time algorithm is proposed to construct k optimal independent spanning trees on a k-dimensional hypercube.
Dong Xiang, Ai Chen, and J. Sun	2005	A new technique was proposed for fault-tolerant routing and multicasting in hypercubes by using a combination of a partial path set-up scheme and local safety information, which is a variant of the pipelined-circuit-switching.	A partial path is set up for fault-tolerant routing, where the header flit is forwarded until a maximal safe subcube containing the current node and the destination is found. Similarly, a partial multicast tree is set-up for multicasting until feasible paths from the leaves of the partial multicast tree to the destinations have been guaranteed.	Backtracking is adopted only for the header along the minimum paths or non-minimum feasible paths if necessary in order to set up a partial feasible multicast tree or a partial feasible path.
M.E. Gomez, N.A. Nordbotten, J. Flich, P. Lopez, A. Robles, J. Duato, T. Skeie, and O. Lysne	2006	This paper presented a new fault-tolerant routing methodology that does not degrade performance in the absence of faults and tolerates a reasonably large number of faults without disabling any healthy node.	In order to avoid faults, for some source-destination pairs, packets are first sent to an intermediate node and then from this node to the destination node. Fully adaptive routing is used along both subpaths.	The methodology assumes a static fault model and the use of a checkpoint/restart mechanism. The proposed fault-tolerant routing methodology is extensively evaluated in terms of fault tolerance, complexity, and performance.

J. S. Yang	2007	The use of multiple independent spanning trees (ISTs) for data broadcasting in networks provides a number of advantages, including the increase of fault-tolerance and bandwidth.	In this paper, based on a simple concept called Hamming distance Latin square, they design a new algorithm for generating *k* ISTs of Q_k	As a result, they showed that the ISTs they constructed are optimal in the sense that both the heights and the average path length of trees are minimized.
O. Sinanoglu, M.H. Karaata, and B. AlBdaiwi	2010	The node-disjoint paths problem deals with finding node-disjoint paths from a source node s to target node t.	Distributed solutions to the node-disjoint paths problem have numerous applications such as secure message transmission, reliable routing, and network survivability.	In this paper, they presented a simple distributed algorithm that is both stabilizing and inherently stabilizing under a realistic model that describes system interfaces and implementation issues in detail to route messages over all shortest node-disjoint paths from one process to another in an n-dimensional hypercube network.
M. Koibuchi, H. Matsutani, H. Amano, D.F. Hsu, and H. Casanova	2012	Modern High Performance Computing (HPC) systems can exploit low-latency topologies of high-radix switches.	They proposed the use of random shortcut topologies, which are generated by augmenting classical topologies with random links. Using graph analysis they find that these topologies, when compared to non-random topologies of the same degree, lead to drastically reduced diameter and average shortest path	Using flit-level discrete event simulation they find that random shortcut topologies achieve throughput comparable to and latency lower than that of existing non-random topologies such as hypercubes and tori.

			length.	
C. N. Lai	2012	Routing functions had been shown effective in constructing node-disjoint paths in hypercube-like networks.	In this paper, by the aid of routing functions, m node-disjoint shortest paths from one source node to other m (not necessarily distinct) destination nodes are constructed in an n-dimensional hypercube.	By taking advantages of the construction procedure, m node-disjoint paths from one source node to other m (not necessarily distinct) destination nodes in an n-dimensional hypercube such that their total length is minimized.
J. Werapun	2012	Reliable data broadcasting on parallel computers can be achieved by applying more than one independent spanning tree (IST).	Using k-IST-based broadcasting from root r on an interconnection network ($N=2^k$) provides k-degree fault tolerance in broadcasting, while construction of optimal height k-ISTs needs more time than that of one IST.	They focused to introduce the more efficient time (O(k)) of preprocessing, based on double pointer jumping over $O(k^2)$ of the HDLS approach.

Clarifications

1. Cost of handling static faults has been said. Why evaluation of the cost of dynamic faults and comparison of their solution with those of others has not been done?

Ans. The dynamic fault behaviour can take place in the absence of the static fault behaviour. So, in the start of the evaluation I considered only static faults. Further evaluation and comparison can be done in future work.

2. Why the fault coverage (for example 100%, 50%) and congestion results for a varying range of faults having different cube sizes are not required?

Ans. I did simulation for various dimensions of hypercube. The fault-tolerant node-to-node routing algorithm can tolerate upto 60% of faulty nodes. The node-to-set disjoint fault-tolerant routing algorithm can tolerate n-1 faulty nodes. The data broadcasting through independent spanning trees (ISTs) can provide n-degree fault tolerance. This method may increase security in hypercube network.

3. Before generalizing the observation do you see any requirement for carrying experiments for various dimensions of hypercube?

Ans. Yes, for carrying out this experiments, no specific system requirement is needed . I start my simulation first on 3-dimensional hypercube, then 4-dimensional and so on. Lastly when I increase dimension at 9, its taking very much time and hanging problem. So, I stop at dimension 8. I did my simulation on 32-bit Windows Operating system with core i7 and 4 GB RAM for various hypercube dimensions.

4. When multiple nodes fail at the same time what way the failure situation is handled.

Ans. All three Algorithms can tolerate multiple node failures at the same time. For dealing the failures, acknowledgement messages (acks) are included in the first algorithm for routing messages over node-disjoint paths in a hypercube network. The second NoSeRo algorithm can tolerate maximum n-1 faulty nodes. The third algorithms tolerate link failures by means of multiple independent spanning trees (ISTs) in a hypercube network. So, all three proposed algorithms can tolerate multiple node/link failures and propagate the message successfully.

5. As a result of configuration, extra-dilation and extra-congestion may take place. A good algorithm must consider these aspects. I could not see any experimentation/discussion in the thesis on these aspects. Is there any reason why those aspects have not been considered?

Ans. The proposed algorithms are reconfigurable and can handle extra-congestion. When the failure occurs, the proposed algorithms are reconfigurable and select the alternative paths. In this way, the algorithms can send the message from source to destination. The proposed algorithms can handle large congestion because hypercube network uses multiple disjoint paths. Broadcasting is non-redundant in the sense that a minimal number of edges, namely 2^n-1, are used. This is useful for minimizing the congestion.

6. A Petri Net is formally defined as PN={P, T, F, W, M_0}. Candidate has used Petri Net to model the scenario. However, the formal definition and the tuple details with respect to the experiment(s) are not seen. In addition it is not clear whether the Petri Net models depend on initial marking or independent of initial markings. Since the analysis approach of Petri Net models shall vary accordingly. Whenever Petri Net approach is used as a standard practice the model requires examination of certain behavioural properties and some kind of established analysis method (such as coverability, matrix equation, reduction or decomposition etc.) needs to be followed. Thus I feel the candidate may explore them for model evaluation and relate them with the problem domain.

Ans. A Petri net is formally defined as a 5-tuple PN={P, T, F, W, M_0}, where

(1) P = {p_1, p_2, . . . , p_m} is a finite set of places;

(2) T = {t_1, t_2, . . . , t_n} is a finite set of transitions, P $\cup$ T $\neq \emptyset$, and P $\cap$ T = $\emptyset$;

(3) F: P×T→N is an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) W: T×P→N is an output function that defines directed arcs from transitions to places; and

(5) M_0: P→N is the initial marking.

A marking in a Petri net is an assignment of tokens to the places of a Petri net. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net.

Tuples in my case: Refer Figure 3.7.

(1) P = {W, X, Y, Z}

(2) T = {send msg, receive acks, transmit msg, transmit acks, receive msg}

(3) F: P×T→N = {1, 2, 3, . . .}

(4) W: T×P→N = {1, 2, 3, . . .}

(5) M_0: P→N = The messages to be sent are positioned at the place send. Each token on this place contains a message number (INT type) and the data contents of the message (DATA type). The place nextsend contains the number of the next message to be sent. Initially this number is 1, and it is updated each time an acknowledgement is received.

Most of the theoretical work is based on the formal definitions and all concepts are illustrated with the graphical representation.

There are two types of properties that can be studies using Petri net models. The types of properties that depend on the initial marking of a Petri net are called behavioural properties and those that are independent of the initial marking are called structural properties. I studies behavioural properties. Some behavioural properties of Petri nets are reachability, boundedness, safeness and liveness. I used most of them for modelling for analysing the results. All of my algorithms send the message from source to destination in the event of failures. This process maintains the reachability property. The multiple disjoint paths provide safeness by providing multiple routes. In the proposed routing algorithms, liveness (or deadlock) freedom is reached by using the dimension order selection in the hypercube.