1. First, we proof it by contradiction in direct graphs. We assume that there is a direct acyclic graph whose number of nodes is finite and doesn’t have a longest path. Then we can find all paths by listing all nodes pairs as source node and destination node. And the distance between any nodes must be finite and no more than N-1, N is the number of nodes, because the path can’t go through any node twice or more for acyclic. It’s contradiction. So any direct acyclic graph has longest path. Second, for unconnected graph, we can divide any unconnected graph into several direct graphs. Each direct graph can use the conclude above. So any unconnected graph has longest way. Proof completed.
2. In a complete binary tree, the shortest path between two nodes is the addition of the paths between each node and their lowest common ancestor node. So we just need calculate the numbered list of the ancestor node for each node, and then we can find the lowest common ancestor(LCA). The part of the list from source/destination number to LCA number is the path between each node and their lowest common ancestor node.
3. To prove that hypercube is optimal in fault tolerance, we just need to prove the fault tolerance of hypercube is N-1, N is the max degree in hypercube. We know that if the dimension of the hypercube is N, then the degree of any node is N. If we can ensure that there are N disjoint path between any nodes in N-cube, then we can make sure that fault tolerance of N-cube is N-1 so that it is optimal. We prove that there are N disjoint path between any nodes in N-cube by mathematics induction.

When N is equal to 1, obviously, there are 1 disjoint path between any nodes.

We assume that there are N disjoint path between any nodes in N-cube.

N+1-cube can be considered as two same N-cube connected. For convenience, we assign the two point as u and v, and the two same N-cube as N1 and N2.

If n and u are in the same part(e.g. N1), then there is N disjoint paths in N1. And there is a link connected u and N2 excepted for all links in the N paths, for the mode of construction of N-cube. The same link can be found around v which is “new” added. Through these links, we can find path N+1 connected u and v whose left links is in N2. So if n and u are in the same part(e.g. N1), there is N+1 disjoint paths in the graph.

If n and u are in different parts(e.g. u in N1, v in N2), then we can find the corresponding nodes in other part(e.g. u’ in N2, v’ in N1). And there are links between nodes and their corresponding nodes, for the mode of construction of N-cube. We can find N-1 disjoint paths between u and v’, because they are in same part N1. And the same to u’ and v. Then we get two disjoint path sets. And the paths between two sets are one-to-one correspondence, in which the path pair have the same number of nodes which is one-to-one correspondence. The nodes in node pair is linked for the mode of construction of N-cube. So the N-1 path pairs can be constructed as N-1 disjoint paths through links in node pairs. We need a path pair as retained paths. Usually the retained path pair can be choose randomly, but if u and v’(or v and u’) is linked we need to choose the links connected u and v’ directly as retained path pair. it can choose any node pair as “bridge”(except v and v’ or u and u’) in other path pairs(except retained path pair) to created a path(one part of this path is in N1 and another part is in N2, they are connected by “bridge”). Except retained path pair, there are N-1 path pair. So we get N-1 disjoint path. And each path in retained path pair uses the link connected v and v’(or u and u’) as “bridge”. So we can get two disjointed paths in retained path pair. From the above, we get N+1 disjointed path(2 in retained path pair, N-1 in other N-1 path pair ) if n and u are in different parts.

In conclusion, if there are N disjoint paths between any nodes in N-cube, there would be N+1 disjoint paths in N+1-cube.

Demonstration finished

**v**

**v’**

**u’**

**u**

The orange paths form a path pair. They use the green link as bridge to create a disjoint path (the dotted blue path) between v and u.

The blue path pair is choose as retained path pair. They use two special grey links (connected v and v’ and u and u’) to create two disjointed paths (dotted red path and dotted black path) between u and v.