SANDIA REPORT
SAND2008-0069
Unlimited Release
Printed January 2008

Network Topology Analysis

David S. Lee and Jeffrey L. Kalb

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2008-0069
Unlimited Release
Printed January 2008

Network Topology Analysis

David S. Lee and Jeffrey L. Kalb
Wireless and Event Sensing Applications, 2664
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0986

Abstract

Emerging high-bandwidth, low-latency network technology has made network-based architectures both feasible and potentially desirable for use in satellite payload architectures. The selection of network topology is a critical component when developing these multi-node or multi-point architectures. This study examines network topologies and their effect on overall network performance. Numerous topologies were reviewed against a number of performance, reliability, and cost metrics. This document identifies a handful of good network topologies for satellite applications and the metrics used to justify them as such. Since often multiple topologies will meet the requirements of the satellite payload architecture under development, the choice of network topology is not easy, and in the end the choice of topology is influenced by both the design characteristics and requirements of the overall system and the experience of the developer.

CONTENTS

EXECUTIVE SUMMARY 7

1 INTRODUCTION 9

2 BACKGROUND INFORMATION 10

3 METRICS FOR TOPOLOGY ANALYSIS 11

4 BRIEF SUMMARY OF CONSIDERED TOPOLOGIES 14

5 DETAILED NARRATIVE OF SELECTED TOPOLOGIES 16
- 5.1 Fully Connected 16
- 5.2 Complete and Incomplete Hypercubes 16
- 5.3 Cube-Connected Cycles 18
- 5.4 Hybrid Fat-Trees 18
- 5.5 2-D Meshes and 2-D Toroids 19
- 5.6 3-D Meshes and 3-D Toroids 20
- 5.7 Entangled Networks 21
- 5.8 Generalized Star 22

6 TOPOLOGY ANALYSIS DATA 23

7 RELIABILITY ANALYSIS 29

8 CONCLUSION 41

9 REFERENCES 42

DISTRIBUTION 43

LIST OF FIGURES

Figure 1. Sample Adjacency Matrix 10
Figure 2. Two Methods for Obtaining Degree From the Adjacency Matrix 12
Figure 3. Fully Connected Network for n=6....................16
Figure 4. Complete Hypercubes (d=1, 2, 3, and 4)....................16
Figure 5. Incomplete Hypercubes (n=7, 9, and 14) 17
Figure 6. Cube-Connected Cycle (d=3)....................18
Figure 7. Hybrid Fat-Tree for d=6 18
Figure 8. 2-D Mesh and 2-D Toroid 19
Figure 9. 3-D Mesh and 3-D Toroid 20
Figure 10. Sample Entangled Networks 21
Figure 11. Generalized Star for d=4 22
Figure 12. Average Path Length 23
Figure 13. Network Diameter 24
Figure 14. Average Node Degree 25
Figure 15. Total Link Count in Network 26
Figure 16. Worst-case Connectivity 27
Figure 17. Algebraic Connectivity....................28
Figure 18. Hypercube Reliability....................30
Figure 19. Entangled Network Reliability, Node Degree of 4 31
Figure 20. Entangled Network Reliability, Node Degree of 6 32
Figure 21. 2-D Toroid Network Reliability, Node Degree of 4....................33
Figure 22. 3-D Toroid Network Reliability, Node Degree of 6....................34
Figure 23. Hybrid Fat-Tree Network Reliability 35
Figure 24. Reliability of Various 16-Node Networks....................36
Figure 25. Reliability of Various 32-Node Networks....................37
Figure 26. Reliability of Various 36-Node Networks....................38
Figure 27. Reliability of Various 64-Node Networks....................39
Figure 28. Reliability of Various 128-Node Networks....................40

EXECUTIVE SUMMARY

Emerging high-bandwidth, low-latency network technology has made network-based architectures both feasible and potentially desirable for use in satellite payload architectures. Network architectures are capable of routing large amounts of traffic with reasonable latency, allowing considerable amounts of data between processing units ("nodes") to be shared. However, care must be exercised when developing these types of architectures. Improper network layout, routing algorithms, or other factors can cause undesirable results ranging from minor inefficiencies (i.e., increased power consumption) to catastrophic failure (i.e., loss of data).

This study examines network topologies and their effect on overall network performance in satellite payload architectures. The topology of a network is defined as the arrangement of nodes and the arrangement of links or interconnections between those nodes. The topology of the system affects many characteristics of the network, including those relating to reliability, performance, complexity, and efficiency. Also, as there is typically not a single ideal topology for all systems being developed, choosing a network topology becomes a matter of balancing benefits and drawbacks to meet the overall requirements of the system.

A number of performance, reliability, and cost metrics were used to evaluate the topologies. The performance metrics include the mean path length (average number of nodes a packet travels through to reach its destination) and network diameter (longest path a packet may traverse). Reliability metrics include node degree (number of links connected to a node), node and link connectivity (number of nodes or links that must fail to segregate part of the network), and discovery of bottlenecks (nodes overloaded beyond their capacity). Algebraic connectivity, a metric derived from linear algebra techniques, is used to evaluate both reliability (indirectly related to the number of cycles in a network) and performance (by mean path length). Cost metrics include the number of links in the system (affecting other characteristics like total power requirements).

A variety of network topologies were investigated for the purpose of this study. Sections 4 and 5 in this paper describe all of the considered topologies. Popular topologies, such as the fully connected topology, 2-D and 3-D meshes, 2-D and 3-D toroids, hypercubes, rings, trees, and stars, were obvious choices for review. Some lesser-known topologies, such as cube-connected cycles, generalized stars, hybrid fat-trees, and entangled networks, were also considered.

The first step in this study was to remove topologies from the list that had obvious deficiencies. As one example, star-based topologies introduce bottlenecks and a potential critical point of failure in their central node. Once this initial down-selection was performed, the remaining topologies to be studied were fully connected, 2-D and 3-D meshes, 2-D and 3-D toroids, generalized stars, complete and incomplete hypercubes, hybrid fat-trees, and entangled networks.

The findings of this study show that node degree has the most impact on creating reliable, high-performance networks. This is perhaps intuitive, as more links in the network provide more paths between nodes. Having additional paths increases reliability, as more redundancy is built into the network, and performance benefits as well due to fewer hops between pairs of nodes. The downside to increasing links in a network is the requirement of additional hardware and the potential for increased power consumption.

When evaluating several networks of similar node degree, entangled networks emerged as the best performer. Entangled networks consistently had the lowest mean path length between nodes and the highest theoretical reliability. Entangled networks had no restrictions on node or link counts and could be built with any size or complexity, making them very versatile. The downside to these topologies is their complex, "entangled" layout, making them difficult to use intuitively.

The toroid topologies were overall good performers with well-defined structures. The 3-D toroids were among the top performers, mostly due to its large node degree of 6. The 2-D toroids fared well, but were outperformed by other topologies such as hypercubes. Despite slightly lower performance and reliability scores, toroid topologies remain a strong choice due to their simple structure and the ability to accommodate networks of many varying node counts.

Hypercube-based networks are good solutions for networks with node counts less than 64, and will consistently outperform 2-D toroids. As node count increases, the node degree of hypercube networks increases as well. This has advantages, such as increased performance and reliability, and disadvantages, such as higher node complexity. Due to this higher node degree, hypercubes outperform 3-D toroids at large (>128) node counts. A significant disadvantage of complete hypercubes is the restrictive node count, as the node count must be a power of two. Incomplete hypercubes alleviate this problem, but can introduce poorly connected nodes, somewhat reducing reliability. However, incomplete hypercubes do gain the versatility of arbitrary node counts.

Ultimately, the choice of network topology is not easy, as often multiple topologies will meet the requirements of the satellite payload architecture under development. The most important detail is matching the requirements of the system with a topology that can perform as required, but also important is the ability of the designer to work with the selected topology. As an example, systems that do not demand maximum performance may choose a simple 2-D toroid to reduce design complexity by its intuitive, easy layout. Alternatively, systems that require maximal performance may choose an entangled network. In the end, the choice of topology is influenced by both the design characteristics and requirements of the overall system and the experience of the developer.

1 INTRODUCTION

The selection of network topology is a critical component when developing multi-node or multi-point system architectures. A good choice of topology will require less power, have less complexity, higher reliability, and will orchestrate network traffic smoothly and quickly between nodes. Conversely, a poor choice of topology will introduce complex and power-hungry logic, reduced performance due to bottlenecks and large distances between nodes, and increase the probability of system failure due to dropped messages and lack of fault tolerance. The purpose of this document is both to provide a handful of network topologies that yield themselves to being good choices to general system applications and to identify the metrics that are used to justify them as such.

Topology selection should begin by identifying the required and desired qualities of the network. These qualities may include a certain level of speed and performance, power usage, wiring or routing complexity, cost, redundancy and reliability, or a combination of these or other factors. Furthermore, individual system requirements may introduce data flow requirements that map to some topologies more readily than others. Only after the requirements of the network have been established can the various topologies be analyzed to determine which will work best for a particular system.

During topology selection, it is important to remember that often there will not be a single ideal topology for most systems. Rather, selecting a topology becomes a matter of balancing and trading off various properties until a reasonable solution is achieved. Another issue to consider is the fact that several topologies may be acceptable given a set of requirements. In this case, selecting among the potential candidates becomes a judgment call on the part of the designer.

As mentioned in the opening paragraph, this document provides a description of fundamental metrics that should be considered when selecting a network topology. This list is not intended to be a complete list of all possible usable metrics; rather, the list provided gives the most common (and debatably the most important) considerations when selecting a topology. Additionally, a detailed analysis of a handful of popular network topologies is provided. A few of these topologies are selected as “preferred” topologies for general use and the justification for those topologies is provided.

2 BACKGROUND INFORMATION

This section of the document will provide some of the assumptions used in this document and will attempt to describe the fundamental elements behind some of the topology metrics and their derivation.

One of the most important assumptions made in this study is that all network links are assumed to be bidirectional. While networks with unidirectional links do exist, the majority of general applications utilizing node-based architecture incorporate bidirectional network links, and as such, this study will follow the majority.

Another important assumption is that the "cost" of each link is equal to that of every other link. Cost is an attribute sometimes applied to network links as a measure of its desirability (i.e., low-cost links are the preferred links, and high-cost links are avoided or used only when necessary). Cost factors in numerous variables beyond the scope of this study (including link types, cable lengths, monetary cost, and link quality, among others), many which may be system-specific, and therefore the cost of each link will be assumed to be equal for this study.

In this document, networks are assumed to consist of two core components: nodes and links. Nodes are responsible for generating information, processing information, or routing information to other nodes. A link connects two nodes together. In some instances, nodes may be referred to as "vertices" and links may be referred to as "edges." This alternate terminology stems from expressions in mathematical graph theory, which is used to analyze some of the properties of these networks.

To analyze networks using graph theory, networks must usually be represented as an adjacency matrix (see Figure 1). The adjacency matrix of a network is an NxN matrix (N = number of nodes) that is populated with a 1 in locations where two nodes are connected and 0 otherwise. The first row and first column represent node 1, and the second row and second column represent node 2, etc. Thus, if node 1 is connected to node 2, and node 2 is connected to node 4, the adjacency matrix will have a 1 in locations (1,2), (2,1), (2,4), and (4,2). This assumes (1,1) is in the first, upper-left element and X and Y increase right and down, respectively.

	Node 1	**Node 2**	**Node 3**	**Node 4**
Node 1	0	1	0	0
Node 2	1	0	0	1
Node 3	0	0	0	0
Node 4	0	1	0	0

Figure 1. Sample Adjacency Matrix

Since every link in our analysis is bidirectional, the adjacency matrix will be symmetric about its diagonal. This is because if node X is connected to node Y, our choice of bidirectional links dictates that node Y is also be connected to node X. This representation of networks as adjacency matrices allows us to utilize linear algebra and graph theory in the analysis of these networks.

3 METRICS FOR TOPOLOGY ANALYSIS

Listed below are the various metrics that were considered when evaluating topology candidates.

- ***Average Path Length:*** The average distance between two nodes in the network over all pairs of distinct nodes. The distance between any two distinct nodes is the shortest path between those two nodes. This can be mathematically represented as:

$$L_{ave} = \frac{\sum_{x=1}^{n-1}\left[\sum_{y=x+1}^{n} d(x,y)\right]}{\sum_{z=1}^{n-1} z}$$ where *d(x, y)* is the distance (number of hops) between nodes *x* and *y*.

Average path length is one of the most important factors when optimizing networks for speed and efficiency. Short average path lengths ensure that messages do not have to travel far to their destination and thus do not remain in the network for long periods of time. Short average path lengths decrease overall network utilization and reduce message latency.

When dealing with many "organized" or structured topologies (especially those following some geometric pattern), some improvement in average path length is often obtained by randomly rewiring a small number of links in the topology. This causes the topology to more resemble a small-world network (a class of random graphs where most nodes are not directly adjacent but can be reached with a small number of hops), which typically have better average path lengths than structured topologies.

- ***Diameter:*** The longest path in the network between two nodes. The diameter of a network is found by recording the shortest paths between all pairs of distinct nodes, and taking the maximum of this set. Utilizing the above representation of distance between nodes, one representation of diameter is:

$$DIAMETER = \left\{\max[d(x,y)] \mid x = \{1,2,3,...,n\}, y = \{1,2,3,...,n\}, x \neq y\right\}.$$

Diameter should be minimized when possible; however, the average path length is usually a more important consideration, since diameter only considers distance between the two farthest nodes. The two factors will typically be related, though a topology with a large diameter will generally have a larger average path length, and a small diameter will generally imply a small average path length.

- ***Node Degree:*** The degree of a node, $d_G(x)$, is equal to the number of links to which that node is connected. To reduce node, network, and routing complexity, a small degree is preferred, as is a fixed, matching degree for all nodes. There is a trade-off between node degree and reliability; more redundant networks will require more links for use as redundant paths in the network, which consequently leads to higher node degree. Be aware that the converse is not true – a high node degree is not necessarily an indicator of reliability or redundancy.

Degree can be determined by observation by visually counting the number of links connected to a particular node when the network is represented pictorially (see Figure 2). Mathematically, degree may be obtained by representing the network via an adjacency matrix, then summing values within a row or column of the adjacency matrix to obtain the degree for a particular node. Alternately, the node degree may be obtained by multiplying the adjacency matrix by itself, using standard linear algebra matrix multiplication [1]. The degree of each node would then be represented along the diagonal of the matrix.

0	1	0	0	← 0+1+0+0 = 1, Node 1 has degree 1
1	0	0	1	← 1+0+0+1 = 2, Node 2 has degree 2
0	0	0	0	← 0+0+0+0 = 0, Node 3 has degree 0
0	1	0	0	← 0+1+0+0 = 1, Node 4 has degree 1

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2. Two Methods for Obtaining Degree From the Adjacency Matrix

Other important factors derived from node degree include average degree, minimum degree, and maximum degree. Average degree is the sum of the degrees of all nodes divided by the total number of nodes in the network. The maximum degree, $\Delta(x)$, is the largest degree over all nodes; the minimum degree, $\delta(x)$, is the smallest degree over all nodes.

- ***Number of Links:*** Increasing the number of links can potentially reduce latency, average path length, network congestion, and increase overall performance, but more links can increase cost and complexity of network wiring and routing due to increased node degree. Optimizing the number of links requires balancing the trade-off between high performance and redundancy vs. lower cost, power, and less inter-node links.

- ***Worst-case Connectivity:*** This is the minimum number of nodes that must fail (node connectivity $\kappa(x)$) or the minimum number of links that must fail (edge connectivity $\lambda(x)$) [2] to cause any additional type of failure in the system. An example of a failure in this case would be the inability for any live node to communicate with any other presently live node due to, say, node failures of all a live node's neighbors.

The connectivity of the network is bounded by the inequality $\kappa(x) \leq \lambda(x) \leq \delta(x)$. Simply put, the worst-case connectivity of the network will be less than or equal to the smallest node degree present in the network (this is because the node with the smallest degree can be isolated by failure of the links that surround it or by failure of the nodes that it is connected to). Also, the number of link failures needed to disconnect the network will be less than or equal to the number of node failures required.

Worst-case connectivity has a lower bound equal to the algebraic connectivity (see below) of the network. Increasing the algebraic connectivity is a good first step to obtaining better connectivity in the network.

- ***Algebraic Connectivity:*** Algebraic connectivity is a metric derived from mathematical graph theory. "Well-connected" graphs will have a large algebraic connectivity [3]. Being "well-connected" implies good average path lengths as well as an abundance of loops to ensure good reliability and overall connectivity. Thus, graphs with high algebraic connectivity generally indicate efficient placement of links with many redundant paths between nodes, as well as good distribution of traffic (depending on the routing algorithm used).

Conversely, graphs with small algebraic connectivity have relatively clean bisections (meaning it is easier to divide these graphs into two halves) [3]. The algebraic connectivity of the graph also indicates a lower bound to node and edge connectivity and expansion. Thus, it is desirable to maximize this value whenever possible.

The algebraic connectivity of a network is determined by finding the eigenvalues of the Laplacian matrix L [4]. The Laplacian matrix L is determined by L = D – A, where D is the degree matrix of the network (a diagonal matrix where the term di,i is the degree of node I) and A is the adjacency matrix representation of the network. Once the matrix L is found, the algebraic connectivity will be equal to the second smallest eigenvalue (λ_2) of this matrix.

Algebraic connectivity $\lambda_2(x)$ relates to node connectivity $\kappa(x)$ and edge connectivity $\lambda(x)$ as shown by Fiedler's inequality: $\lambda_2(x) \leq \kappa(x) \leq \lambda(x)$.

- ***Scalability:*** The ease in which the number of nodes within a network may be changed. A high scalability is desired. Some topologies require a fixed number of nodes in a specific structure to operate; others may allow an arbitrary number of nodes. This is purely dependent on the topology chosen for use in the network.

- ***Routing Complexity:*** The complexity of routing algorithms must be considered, especially when in environments where the routing table must be dynamically generated due to node or link failure. Some topologies yield themselves to easy mathematically based routing algorithms (e.g., hypercube or toroidal layouts) while others may rely on more general routing algorithms and techniques [such as the industry-standard open shortest path first (OSPF) algorithm].

- ***Bottlenecks or Points-of-Failure:*** The presence of bottlenecks can introduce a slew of other problems, including potential for network overload (causing packet delays or drops), increased latency, and the introduction of certain nodes or links becoming critical points-of-failure. Networks with short average path lengths may lose the advantage of their short transmission distances if bottlenecks exist, restricting traffic flow between nodes. Note that bottlenecks may exist as either nodes or links.

Failures of bottlenecks will severely, and sometimes catastrophically, impact the system. From a reliability standpoint, bottlenecks should be avoided as the failure of a single link or node can cause moderate to severe impact. From a performance standpoint, bottlenecks considerably impair network performance. Bottleneck nodes and links should be avoided whenever possible.

4 BRIEF SUMMARY OF CONSIDERED TOPOLOGIES

Most of the considered topologies or general topology classes are briefly listed below, along with key factors. The topologies considered for further numerical analysis were fully connected, hypercubes, hybrid fat-trees, generalized star, cube-connected cycles, 2-D meshes and toroids, 3-D meshes and toroids, and entangled networks. In the list below, these topologies are highlighted in blue. Further detail of these topologies follows in the next section.

The best topologies selected for potential implementation and prototyping are (listed from most desired to least desired): Entangled networks, 3-D toroids, hypercubes, and 2-D toroids.

- ***Fully Connected:*** Best performer, but most costly. This topology is restricted by the large number of links required, equal to $\frac{n(n-1)}{2} \sim n^2$, as well as the large number of ports required per node, equal to *n-1*. Thus, this topology is ideal for small networks, but not practical for high node counts due to tremendous increases in required hardware as node count increases.

- ***Hypercubes:*** A good, reliable performer. Limited choice of network sizes as this topology requires 2d nodes, but the extension to incomplete hypercubes somewhat alleviates this problem [5]. This topology begins to become somewhat complex and costly as node count increases, and as such may be preferred for low-to-medium node counts. However, on the plus side, overall reliability correspondingly increases as well.

- ***Incomplete Hypercubes:*** An extension to hypercubes that allows arbitrary node count with hypercube-like performance. Performance is only slightly lower than the complete hypercube, but reliability of the incomplete hypercube can vary substantially.

- ***Tree Topologies (including binary trees and pyramid-style topologies):*** Tree topologies are poor as a general network topology. A severe bottleneck and point-of-failure is present at the root of the tree, and these networks have very large diameter and average path lengths. Fat-trees attempt to lessen the bottlenecking at the root of the tree by increasing bandwidth at higher levels; however, reliability is still a concern.

- ***Hybrid Fat-Tree:*** An extension to a fat-tree topology that interconnects the sides of a tree topology at different points [6], mainly to increase reliability and decrease average path length. However, this topology has varying node degree (implying increased node complexity) and bottleneck issues are introduced at key nodes.

- ***Banyan Networks (including, among others, butterfly and omega networks):*** These topologies are multistage switched networks, where switching hardware is independent of the nodes. This style of network is not desired in our environment.

- ***Star Topologies:*** Poor reliability due to point-of-failure at the node located in the center of the star. A huge bottleneck is present as well. In fact, a star topology can be generalized down to a type of tree topology, and thus has similar issues.

- ***Generalized Star:*** Good in areas of performance and node complexity. The generalized star design scales well with increasing performance and reliability as node count increases. The fixed structure, though, only allows for specific node counts in the network: 6, 24, 120, 720, etc. There are no provisions for networks of 25-119 nodes, 121-719 nodes, etc.

- ***Cube-Connected Cycles:*** Based on the hypercube where each hypercube node is replaced with a small ring. All nodes have a fixed degree of 3, despite network size. The low node degree can be an advantage if port counts at each node are limited, but performance suffers. Network node counts are also fixed at specific node counts of 8, 24, 64, 160, etc. (similar to generalized star's node count constraints).

- ***Ring Networks:*** Ring networks distribute traffic fairly, but not efficiently. A lack of links between nodes limits their ability in areas of performance and reliability. However, cross-connecting multiple rings can yield good results. The 2-D toroid is based on this principle.

- ***2-D Mesh:*** Simple layout, but poor traffic distribution and differing node degrees between core nodes and edge nodes.

- ***3-D Mesh:*** Similar to the 2-D mesh, but with larger metrics in general (node degree, link counts, etc.). Does perform somewhat better and is more reliable than the 2-D mesh.

- ***2-D Toroid:*** Similar to the 2-D mesh, but with edge nodes connected back around to each other. This nicely fixes the node degree at 4 and reduces overall complexity. Easy routing algorithms may be utilized. Moderate performance and reliability.

- ***3-D Toroid:*** Similar to the 3-D mesh, but with edge nodes along all three dimensions connected back to each other. Node degree is nicely fixed at 6. Excellent performer and highly reliable.

- ***Entangled Networks:*** A non-standard topology constructed by optimizing a network with graph theory fundamentals [4]. The goal is to maximize the algebraic connectivity of the final network. Node count and degree are fixed at any desired value. A very good performer and very reliable. However, due to the random nature of the optimization process (a simulated annealing algorithm), it may be difficult to obtain consistently identical networks at moderate-to-high node counts. Despite the variance in optimized topologies, all entangled networks generally perform similarly.

- ***Ramanujan/Expander Graph Topologies:*** Topology formed with fundamentals of graph theory principles [4]. Similar to entangled networks; however, node counts are at various fixed intervals. Not many opportunities to form networks of smaller size due to the node count constraint.

- ***Bus-based topologies:*** The use of a common data line is impractical at higher node counts or higher speeds. Also, these networks are focused more on broadcast-type traffic, versus point-to-point traffic.

5 DETAILED NARRATIVE OF SELECTED TOPOLOGIES

Following are descriptions of topologies that were selected for further detailed study. Each topology is depicted pictorially and outlined, and the advantages and disadvantages of each structure are given.

5.1 Fully Connected

Fully connected networks are constructed by wiring every node to every other node present in the network (see Figure 3). For obvious reasons, these networks are the ideal topology when considering overall speed, diameter, routing complexities, reliability, and ease of construction. However, they require a massive number of links; hence, cost and power consumption are very high. Also, the port count at each node is large, equal to *n-1*. Thus, this topology is generally only good for low node counts.

Advantages: Lowest latency, diameter, and average path length. Best possible reliability, easy to construct, and easy routing algorithm.

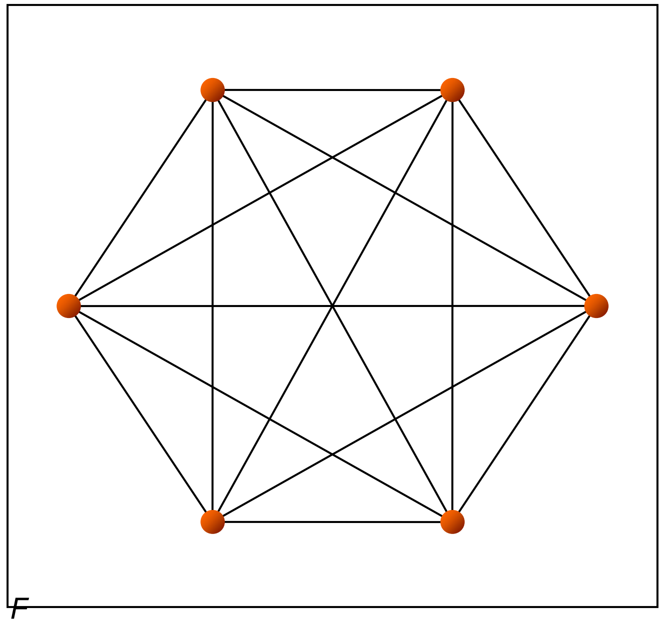

Figure 3. Fully Connected Network for n=6

Disadvantages: Enormous link and port count. Cost and power requirements are very high. May require excessive wiring, which can introduce physical layer challenges. Requires $\frac{n(n-1)}{2} \sim n^2$ *links, and n-1 ports per node.*

5.2 Complete and Incomplete Hypercubes

An overall good performer, hypercubes are very reliable and offer good performance. Complete hypercubes have fixed size of 2^d (i.e., 4, 8, 16, 32, 64, etc, nodes) but an extension to this structure (incomplete hypercubes) allows for arbitrary size.

Hypercubes are constructed by beginning with two interconnected nodes (a 1-D hypercube) (see Figure 4). If more nodes are required, the structure is duplicated and interconnected by adding links between the original and the duplicated nodes.

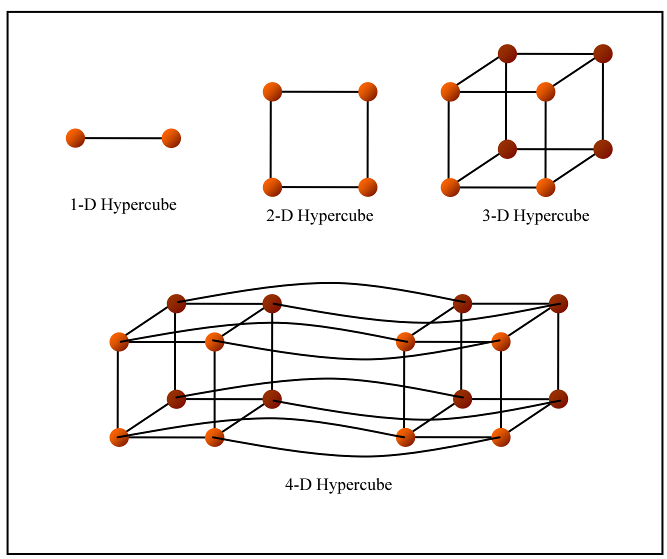

Figure 4. Complete Hypercubes (d=1, 2, 3, and 4)

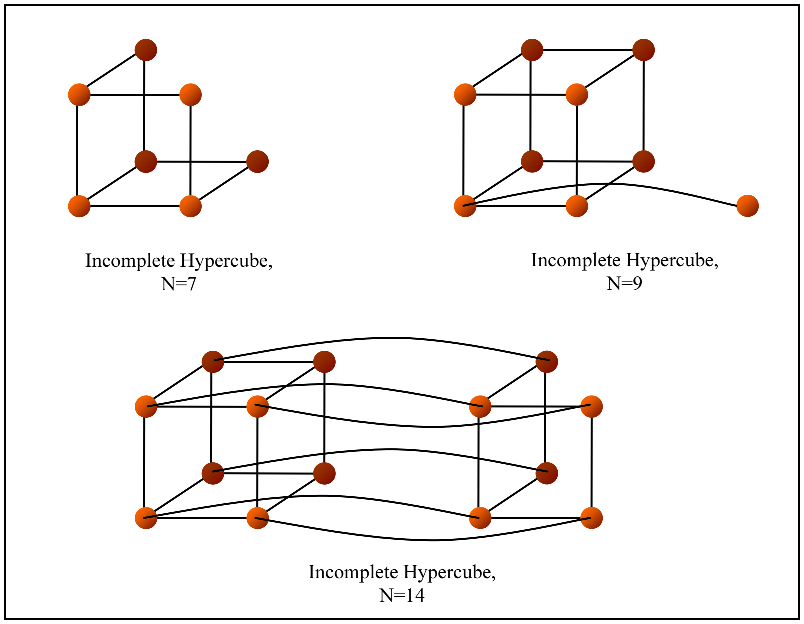

Figure 5. Incomplete Hypercubes (n=7, 9, and 14)

Node degree grows quickly, making node complexity an issue for larger networks. Link count also exceeds other topologies for similar node counts.

Hypercubes offer excellent connectivity, especially at large node counts. This topology also has a high probability of withstanding random link/node failures.

Incomplete hypercubes relax the requirements of complete hypercubes to allow for structures that resemble hypercubes without the strict node count constraint. They generally perform about as well as complete hypercubes. Also, easy routing algorithms have been developed that work under both complete and incomplete hypercube topologies [5].

Reliability in incomplete hypercubes, however, is somewhat unpredictable. Incomplete hypercubes slightly larger than 2^d will introduce points of failure and decrease reliability, while hypercubes slightly smaller than 2^d will perform almost as reliably as a complete hypercube of size 2^d.

As an example, look at Figure 5 for an incomplete hypercube of N=9 – if the link between nodes 1 and 9 were to fail, or if node 1 were to itself fail, then node 9 would be isolated from the rest of the network. Thus, the failure of one node causes two nodes to "fail" (or, the failure of a single link would additionally cause one node to "fail"). However, for networks closer in size to 2^d (such as the above case where N=7), the reliability of the hypercube is closer to that of a hypercube with d=3 (N=8).

The problem of variable reliability could be solved somewhat by adding links to the "incomplete" portion of the topology. However, this is an "impure" solution and adding these additional links makes the resulting topology deviate from the hypercube model.

Advantages: Complete hypercubes offer good to very good performance, very good fault tolerance, and easy algorithms exist for routing traffic through hypercubes. Incomplete hypercubes have performance at near-complete-hypercube level and can have an arbitrary number of nodes. Easy routing algorithms also exist for incomplete hypercubes.

Disadvantages: Node degree grows quickly as the number of nodes increase. Incomplete hypercubes have unpredictable reliability. Link counts grow faster than other topologies, perhaps indicating that a more optimal topology could be used at higher node/link counts.

5.3 Cube-Connected Cycles

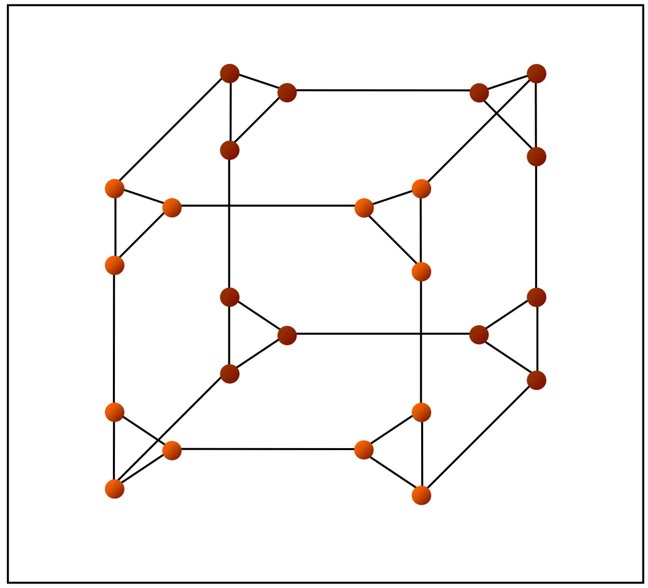

Figure 6. Cube-Connected Cycle (d=3)
(Note the similarity to the hypercube, except each hypercube node is replaced with a ring of size d)

Cube-connected cycles (CCC) are structures based on hypercubes (see Figure 6). Given a hypercube of dimension *d*, each hypercube node is replaced with a ring of size *d*. This causes the node degree of all nodes to be fixed at 3 regardless of network size! This structure excels when the port count at each node is limited. However, the low node degree yields longer average path lengths and diameters than other more highly connected topologies.

The abundance of loops and alternate paths require a robust routing algorithm, but supports good load-balancing, a lack of bottlenecks, and an abundance of different paths between nodes. A major disadvantage is the fixed number of nodes required for this topology, which is equal to $d*2^d$ nodes (i.e., 8, 24, 64, 160, etc., nodes)

Advantages: Fixed node degree of 3 for all nodes. Good load-balancing and redundancy. Low overall link count due to low node degree.

Disadvantages: Does not perform as well as more highly connected topologies. Larger average path length and diameter. Certain link failures (between each "loop") can have a more significant impact on performance than others.

5.4 Hybrid Fat-Trees

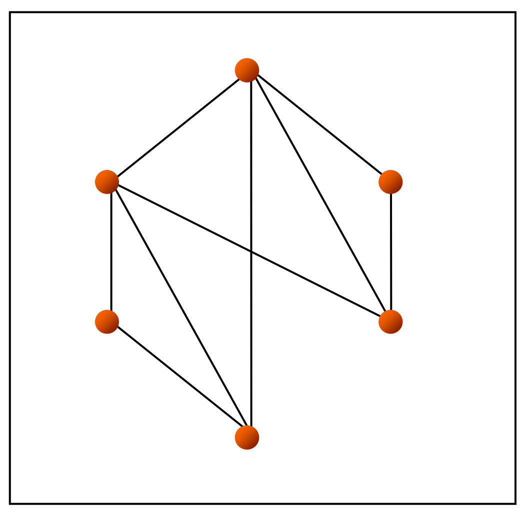

Figure 7. Hybrid Fat-Tree for d=6

The hybrid fat-tree was not a strong contender in this analysis. This topology was mainly included to demonstrate the performance of a tree-style topology (see Figure 7).

A hybrid fat-tree is an extension to the fat-tree topology [6]. Hybrid fat-trees perform well when minimizing average path length and network diameter. One advantage to this design is typically about half of the nodes in this design will have a fixed degree of 2. However, other nodes have large degree, and these nodes introduce bottlenecks into the network as well as significant points-of-failure. Thus, this topology is not a candidate for further testing due to these issues.

Advantages: Low fixed degree of 2 for many nodes.

Disadvantages: Network is easily bisected. Variable node degree and potential bottleneck nodes exist.

5.5 2-D Meshes and 2-D Toroids

2-D meshes are one of the easiest topologies to visualize – nodes are connected in a “grid” fashion (see Figure 8). The simple layout also allows many problems to map easily to the structure of this network [7].

2-D meshes have unequal node degree. The node degree in the corners is 2, around the edges is 3, and in the center is 4. Also, traffic distribution is unequal among nodes. Nodes in the center of the mesh will carry the majority of traffic (leading to possible bottlenecks), and nodes around the edges will carry mostly their own traffic and little of others.

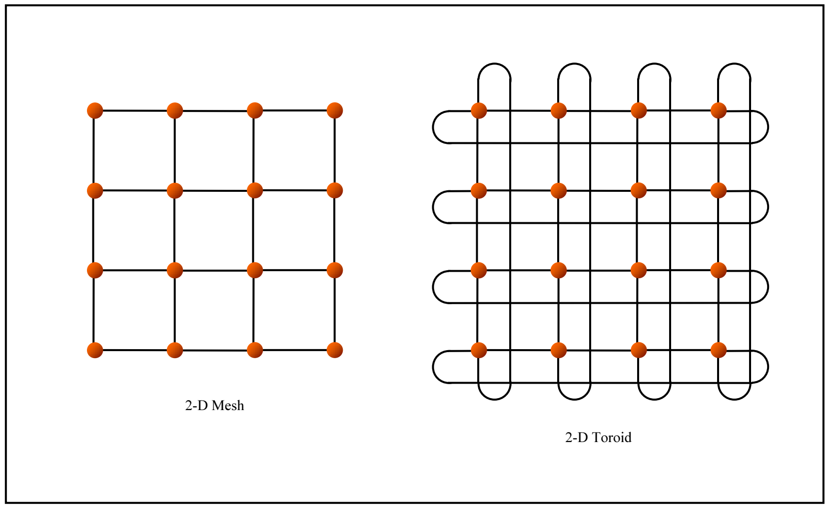

Figure 8. 2-D Mesh and 2-D Toroid

For reliability, 2-D meshes, in the general case, offer many redundant paths between nodes and can probably withstand a fair number of random failures. However, in the worst case, two link failures can isolate a corner node, as could two node failures.

2-D toroids improve on the design of the 2-D mesh by connecting left-edge nodes to right-edge nodes and top-edge nodes to bottom-edge nodes (geometrically, imagine a cylinder where the top is bent around and connected back to the bottom – this is a torus). The benefit to this modification is that each node now has a fixed degree of 4, and reliability and performance improves substantially.

2-D toroids can withstand a high number of general (random) link and/or node failures. In the worst case, four link failures or four node failures could cause the isolation of a valid working node. There are an abundant number of loops and alternate paths in the network to keep this topology connected.

Regarding performance, the addition of links between opposing edges significantly reduces the overall network diameter and average path length. The additional links also distribute the traffic evenly among all nodes, removing the bottlenecks that were present in the center of the 2-D mesh topology. However, the topology could use a few additional links (or rewired links) that jump a few nodes to help improve average path length.

Advantages (2-D Mesh): Simple layout and easy to construct. Problems map well to this topology. Many redundant paths exist between most nodes.

Disadvantages (2-D Mesh): Bottlenecks may exist in the center of the network. Corner, edge, and center nodes all have different degree. Bad worst-case failure rate. One of the easier topologies to bisect.

Advantages (2-D Toroid): Reliable topology with reasonable performance. Fixed node degree of 4 for all nodes.

Disadvantages (2-D Toroid): Link placement is not quite as efficient as some other topologies with the same number of links, and thus average path length and diameter are higher than other topologies with the same link count.

5.6 3-D Meshes and 3-D Toroids

3-D meshes and 3-D toroids are similar to 2-D meshes and toroids, except the 3-D mesh/toroid is expanded along the Z-axis to provide another dimensional layer of nodes. In the case of the 3-D toroid, the topmost nodes (along the new Z-axis) are connected to the bottommost nodes (see Figure 9).

The advantages and disadvantages of 3-D meshes and toroids are similar to those of the 2-D meshes and toroids, but are amplified proportionally with the height of the added dimension. However, due to the added dimension, there is added redundancy and even more paths and loops within the network.

3-D toroids have a fixed degree of 6 for all nodes.

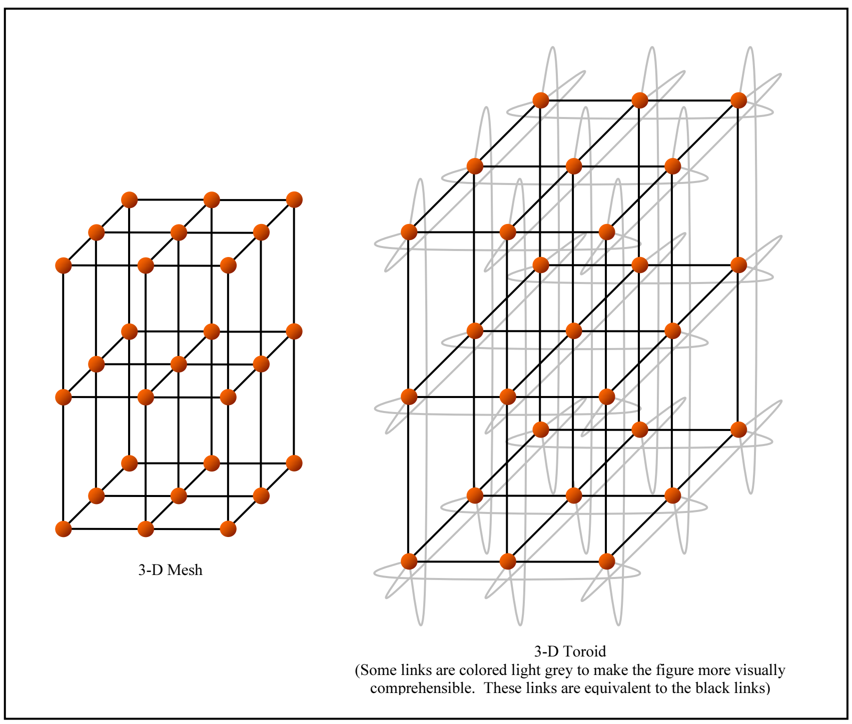

Figure 9. 3-D Mesh and 3-D Toroid

Advantages (3-D Mesh): Simple layout and easy to construct. Many redundant paths exist between most nodes.

Disadvantages (3-D Mesh): Bottlenecks may exist in the center of the network. Corner, edge, and center nodes all have different degree. Fair worst-case failure rate. One of the easier topologies to bisect.

Advantages (3-D Toroid): Extremely reliable topology with reasonable performance. Fixed node degree of 6 for all nodes.

Disadvantages (3-D Toroid): Link placement is not quite as efficient as some other topologies with the same number of links. Thus average path length and diameter are higher than other topologies with the same link count.

5.7 Entangled Networks

Entangled networks are a class of networks that attempt to maximize the algebraic connectivity of the topology. These networks can be constructed with explicit complex mathematical methods or approximated by a repetitive random optimization algorithm (which is the method chosen for this analysis) (see Figure 10).

The repetitive random optimization algorithm takes a topology and randomly rewires two links. The algebraic connectivity of the new rewired graph is calculated. If the new graph has a larger algebraic connectivity the rewiring is made permanent. However, if the rewiring does not result in a higher algebraic connectivity value, the original graph is restored. This rewiring is repeated many times until the topology shows no further improvement to rewiring.

These graphs are considered to be optimal with respect to algebraic connectivity [6]. Since algebraic connectivity is related to average path length and diameter as well as failure rate and reliability, maximizing this value for a given node degree should yield the most (or one of the most) reliable, best performing topology given the available resources.

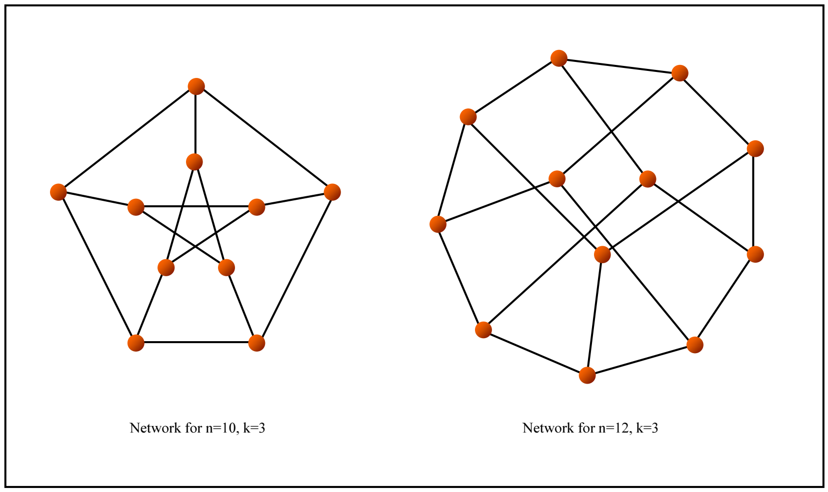

Figure 10. Sample Entangled Networks

Advantages: Theoretically these topologies are optimal in balancing average path length and connectivity for given node count and node degree. The number of nodes as well as the degree of nodes is completely variable and may be set to any value prior to the optimization process.

Disadvantages: Robust routing algorithm is required. Due to the random nature of the optimization process, as well as the possibility of encountering local minima/maxima in this process, a truly optimal network cannot be guaranteed unless one is explicitly constructed (note, too, explicit construction introduces a fair number of other constraints). For larger networks, optimization may take some time.

5.8 Generalized Star

The generalized star topology is a topology that attempts to wire the network together in such a way that each node is the center of a small star (see Figure 11). Given a dimension *d*, these networks will have *d!* nodes and each node will have a fixed degree of *d-1*.

This topology is good in areas of performance and complexity. Generalized stars maintain a low diameter while utilizing a low number of links. Reliability scales with the dimension of the overall network.

This topology is highly specific in structure and thus can only accommodate systems with *d!* (factorial) nodes (i.e., 6, 24, 120, 720, etc., nodes). The large distance between potential node counts does not make this topology feasible for our purposes.

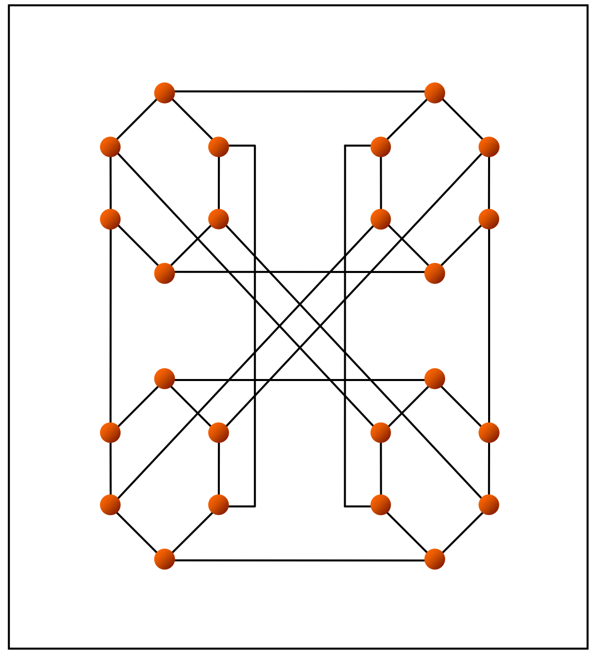

Figure 11. Generalized Star for d=4

Advantages: Low link count, efficient use of links yielding good diameter. Reliability is good and scales with the number of nodes present.

Disadvantages: Very specific in structure, and specific requirements on the number of nodes makes this topology a poor candidate for most systems.

6 TOPOLOGY ANALYSIS DATA

Graphs (Figures 12 through 17) are included on the following pages using the above detailed topologies.

A small narrative follows each graph. In these narratives, the fully connected topology is often ignored since it is in a somewhat different category than the rest of the topologies.

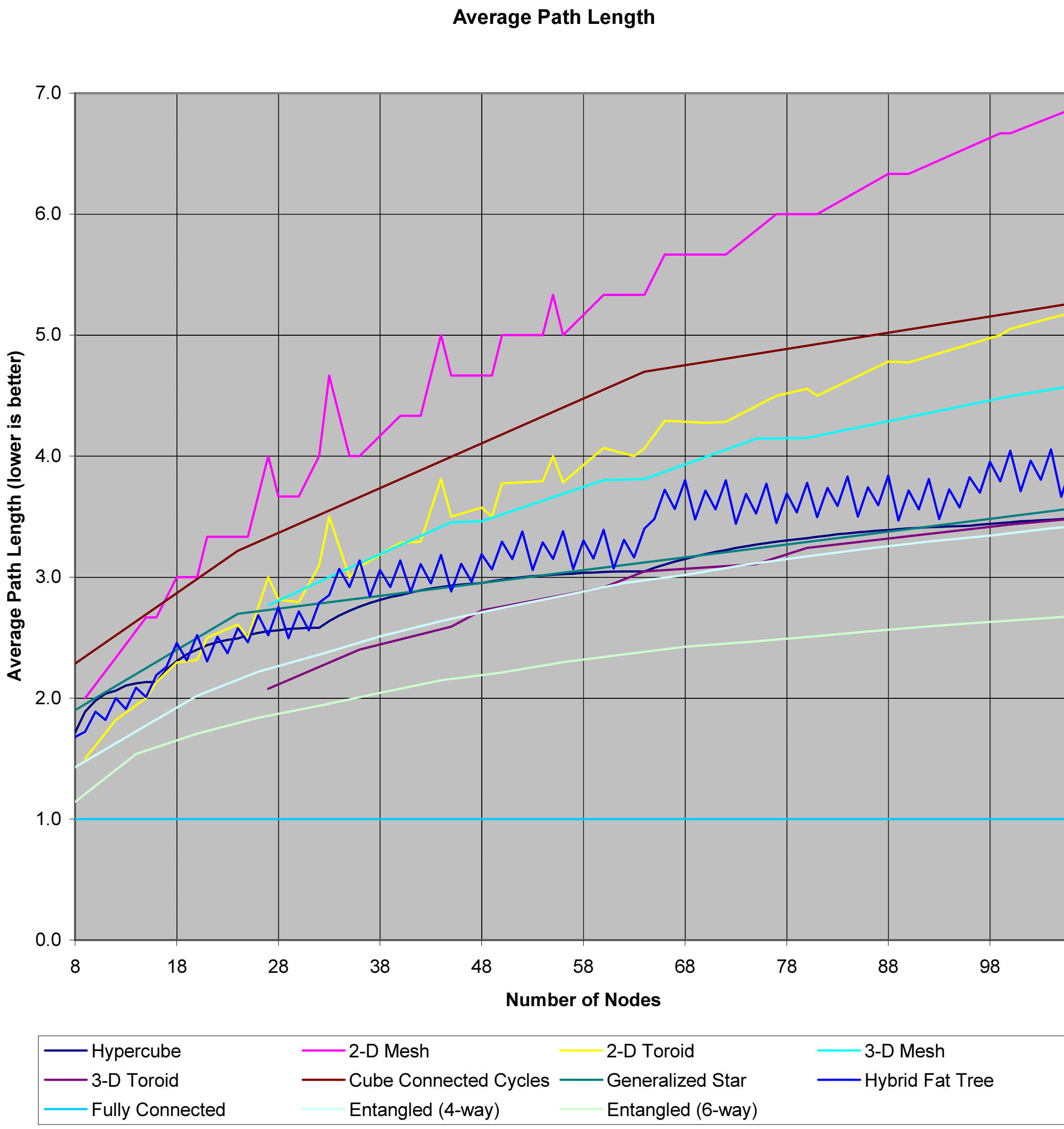

Figure 12. Average Path Length

Shorter average path lengths are preferred, leading to decreased latency and decreased network traffic (due to the reduction in traffic needing to be routed long distances). The optimized entangled networks have the shortest average distance between nodes. The 3-D toroid and hypercube topologies are next, followed by generalized star and hybrid fat-trees. The remainder of the topologies (2-D and 3-D meshes, CCC, and 2-D toroid) are somewhat inefficient and thus have higher average distances between nodes.

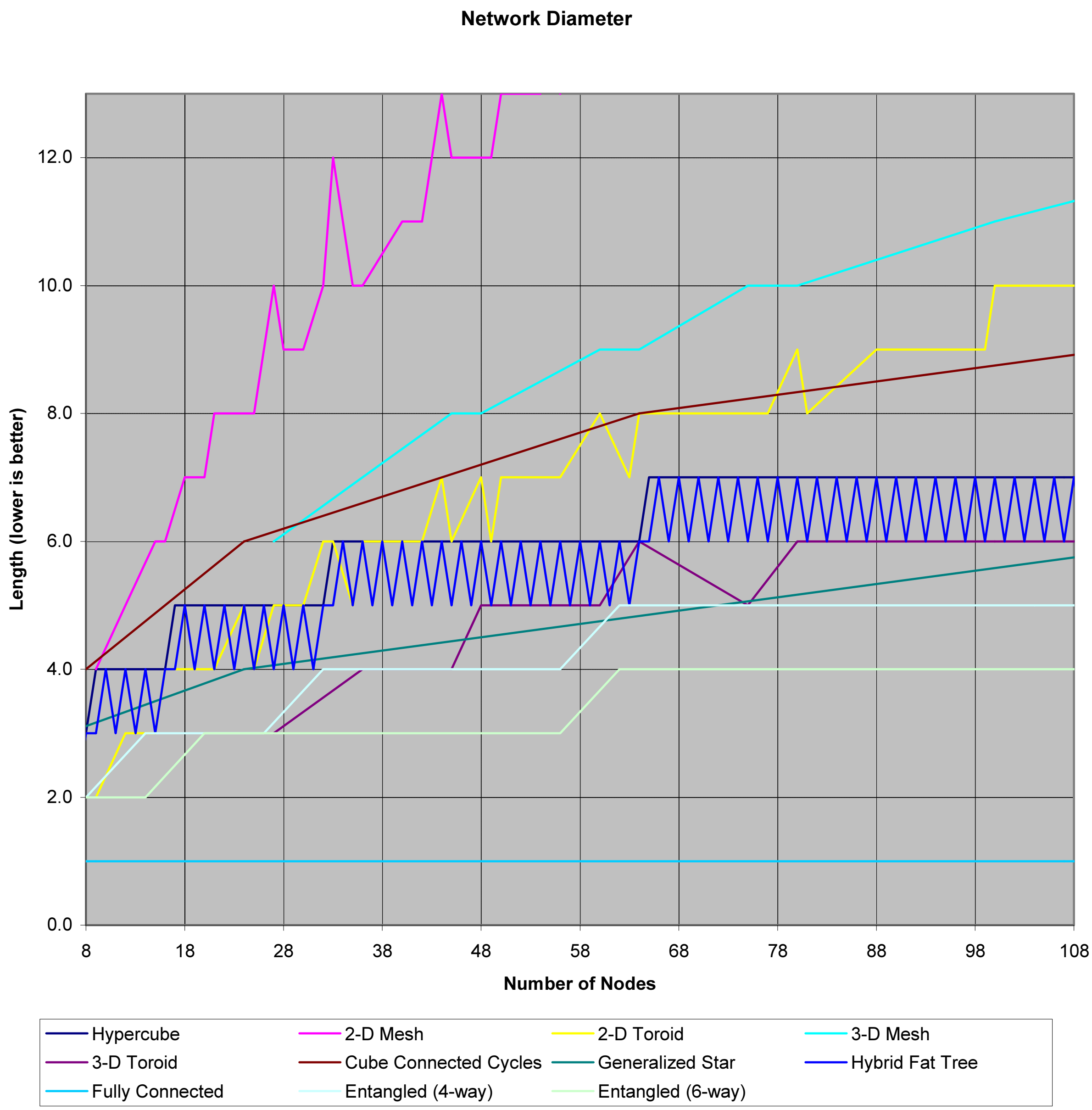

Figure 13. Network Diameter

Network diameter should be minimized whenever possible to reduce distance between nodes. The best network diameter is led again by the optimized entangled networks and 3-D toroids. Now, though, generalized star topologies are similar in performance to the 3-D toroids, as are hybrid fat-trees. Meshes have the worst diameter.

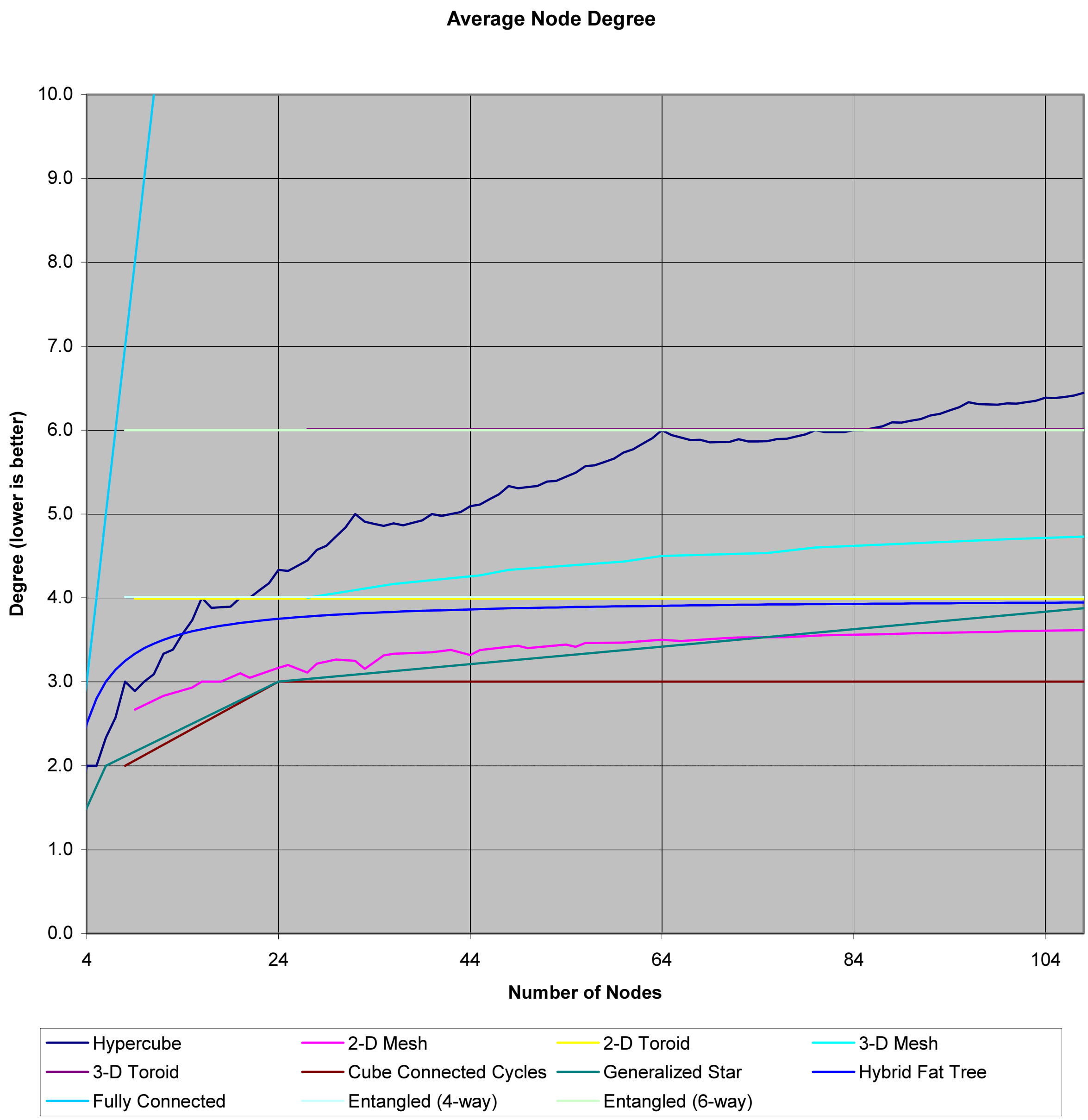

Figure 14. Average Node Degree

The CCC lead the way in average node degree with their fixed degree of 3, making them a strong contender when port counts are limited at each node. Other desirable topologies are the entangled networks and the toroids, all of which have fixed degree.

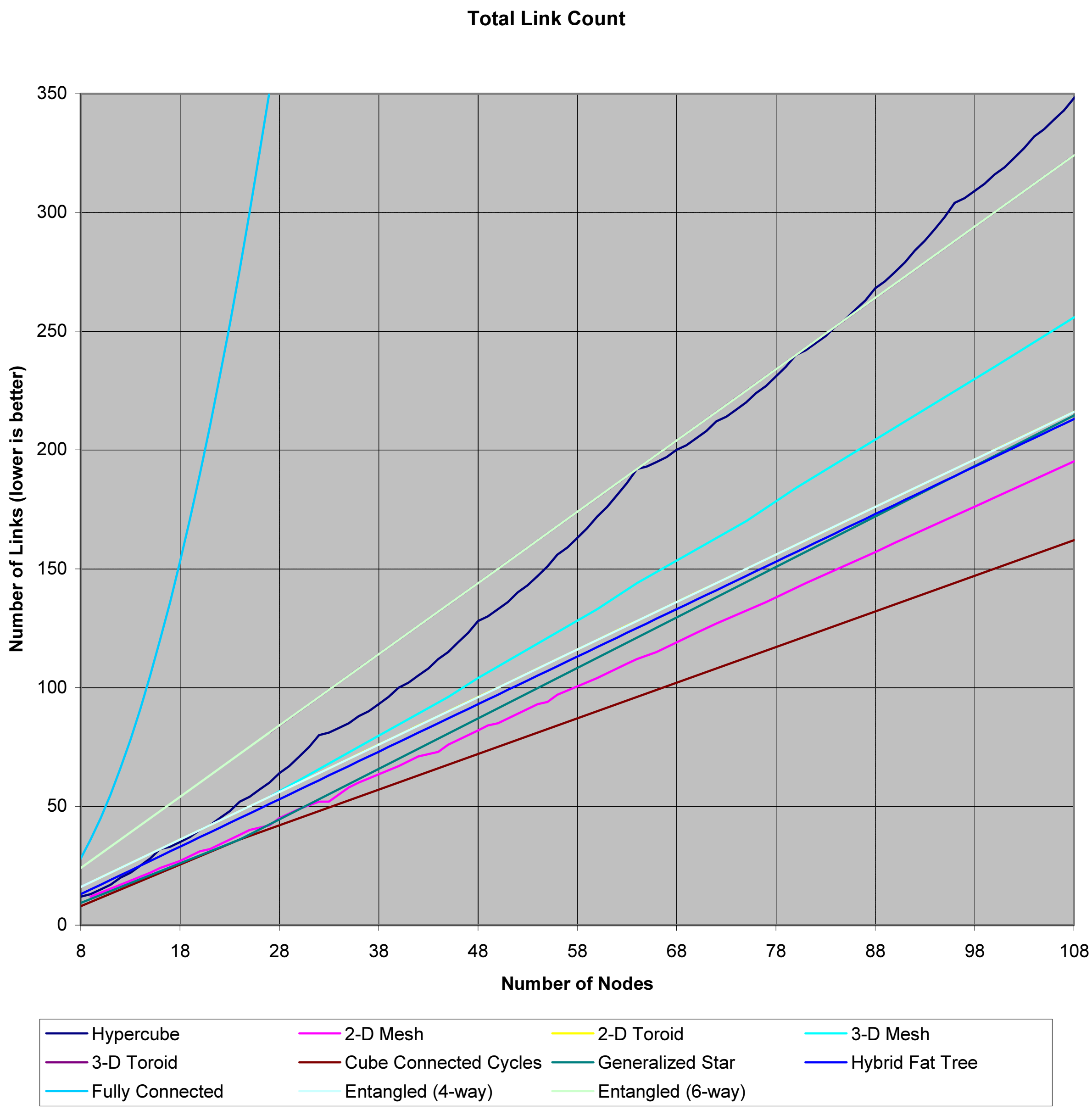

Figure 15. Total Link Count in Network

The number of links can be linked to overall cost and power usage, and can contribute somewhat to the complexity of the topology. Most important to realize is the slope of the line, which indicates the rate at which links need to be added as the network grows. The less steep the slope, the better. Again, the entangled networks perform best, especially when compared to other topologies having the same or similar node degree. Hypercubes are among the worst – their link count grows at the fastest rate as node count increases.

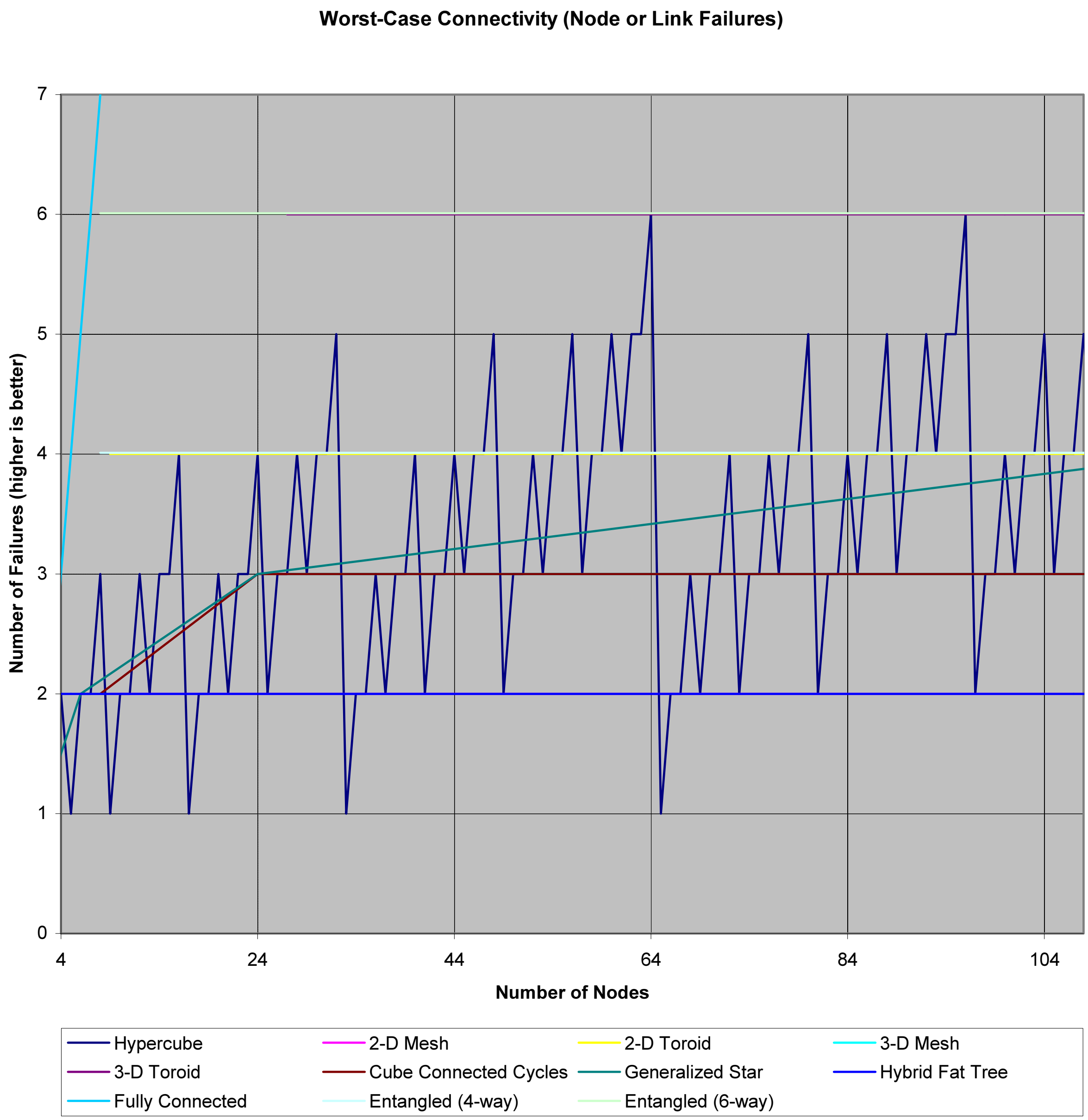

Figure 16. Worst-case Connectivity

The worst-case connectivity is a small part of the overall reliability of the topology. This value is upper-limited by the lowest degree of all nodes present in each topology. Thus, incomplete hypercubes perform poorly here, as topologies with slightly greater than 2^n nodes will have low node degree at their lone incomplete segments and thus be limited in their worst-case connectivity by these segments. Otherwise, complete hypercubes perform relatively well, as do entangled networks and toroids.

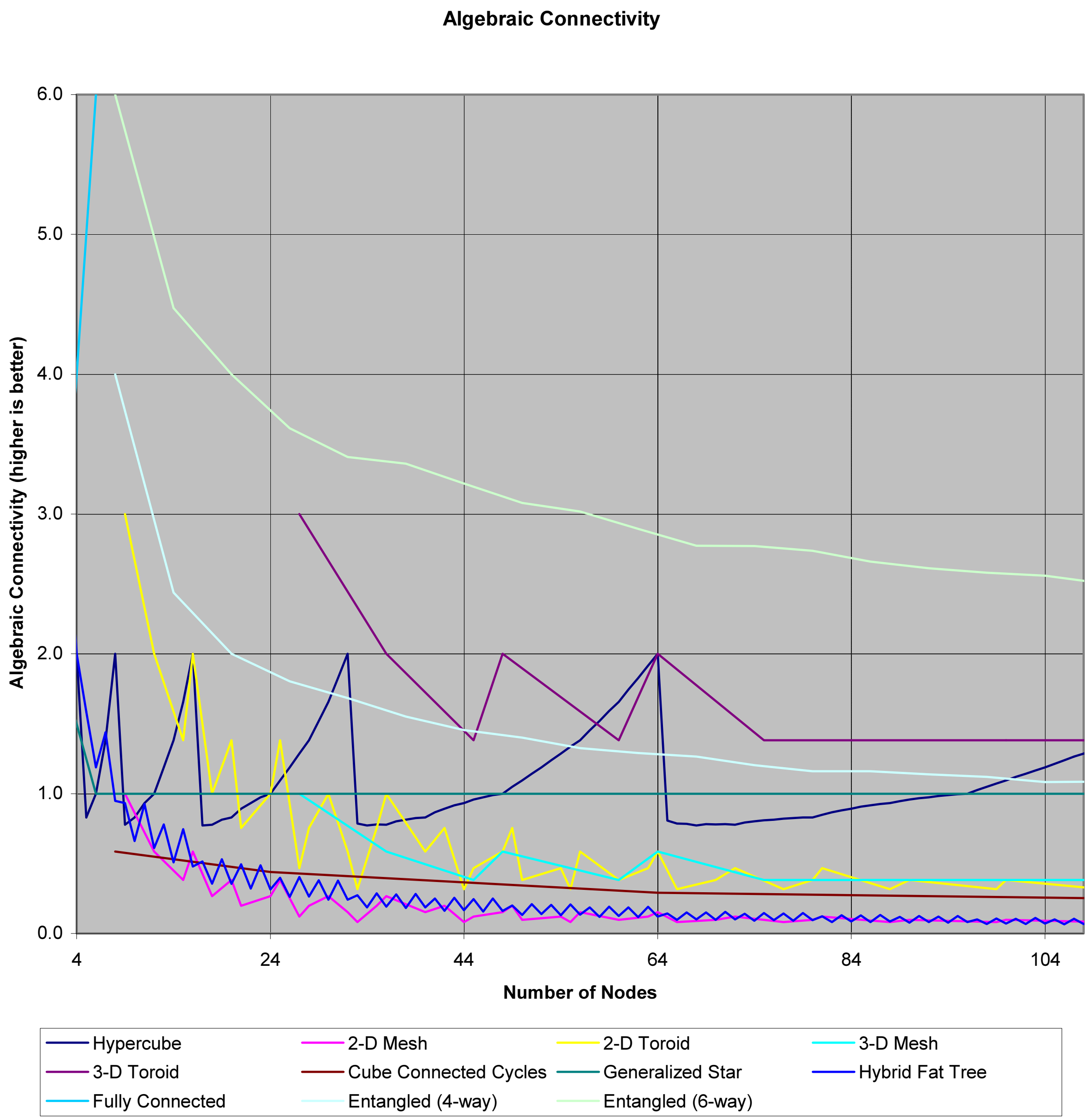

Figure 17. Algebraic Connectivity

The algebraic connectivity is greatest for the entangled networks, when compared to other topologies of similar degree. This should be the case, since these networks strive to specifically optimize this value.

3-D toroids and hypercubes (especially complete hypercubes) are the better performers, with generalized star topologies at the middle of the road. Meshes and hybrid fat-trees are easy to bisect, and thus have low algebraic connectivity.

7 RELIABILITY ANALYSIS

To further expand upon the reliability of the leading topology contenders, a simulation was run to more closely mimic the real-world behavior of these topologies under different rates of node failure. Specifically, the simulation tested the ability of live nodes to communicate with each other despite failures elsewhere in the system. This simulation is similar to the experiments of Abachi and Walker [8] for testing topology reliability.

The simulation generated the top four of the studied topologies under specific node counts. At each node count, nodes were failed throughout the system with specific probabilities of failure under uniform distribution. Using this partially failed network, 500 packets were transmitted by live nodes to other live nodes in the topology. The number of successful transmissions determined the reliability of the system at that particular node count and failure rate. To ensure a good statistical sampling, each combination of topology, node count, and node failure rate and distribution was tested 500 separate times (each with 500 packets) and the results combined to form the final figures.

The results generated are shown in two different ways: (1) the effect of node count on overall system reliability given a particular topology, and (2) the comparison of different topologies on system reliability given identical (or relatively close) node counts.

The reliability of the hypercube generally increases with increasing node count (due to the increase in node degree as node count increases) (see Figure 18). The exception is in the cases when incomplete hypercubes are considered. As an example, compare 64- and 96-node hypercubes. The incomplete 96-node hypercube has 50% more nodes, yet its overall system failure rate is approximately equal to the 64-node complete hypercube. Hypercubes of smaller sizes do not tolerate failures well, mostly due to their low node degree.

Entangled networks are very capable of handling a fair number of failures (see Figures 19 and 20). The network is able to route almost all packets between live nodes despite node failure rates up to almost 40%! Node count has very little bearing on reliability until node reliability falls under 40%. After this point, the reliability of entangled networks actually decreases a bit with increasing node count. As in the hypercube case, increasing node degree in these networks significantly increases reliability.

The prime competitor to the 2-D toroid is the entangled network with degree 4 (see Figure 21). The entangled network does perform slightly better when comparing similar node counts.

The prime competitor to the 3-D toroid is the entangled network with degree 6 (see Figure 22). Both networks are extremely reliable, even under higher failure rates. Even with 50% of nodes failing, this network was still able to transport messages between live nodes approximately 95% of the time, for any of the tested node counts.

For comparison sake, the reliability model of the hybrid fat-tree was simulated and plotted. Figure 23 shows the failure rate of less reliable, less well-connected topologies with low algebraic connectivity. Any number of failures significantly affects the system.

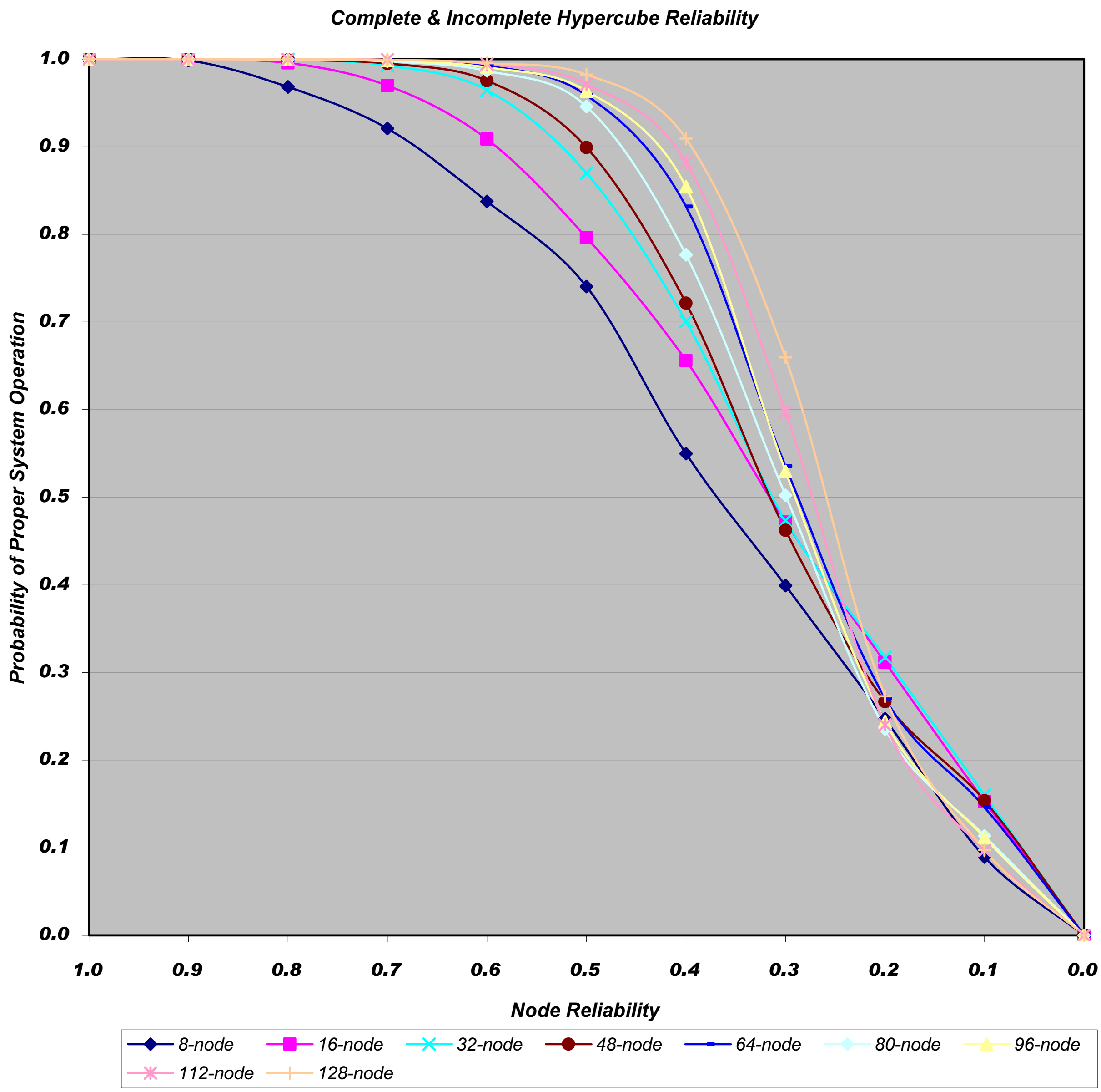

Figure 18. Hypercube Reliability

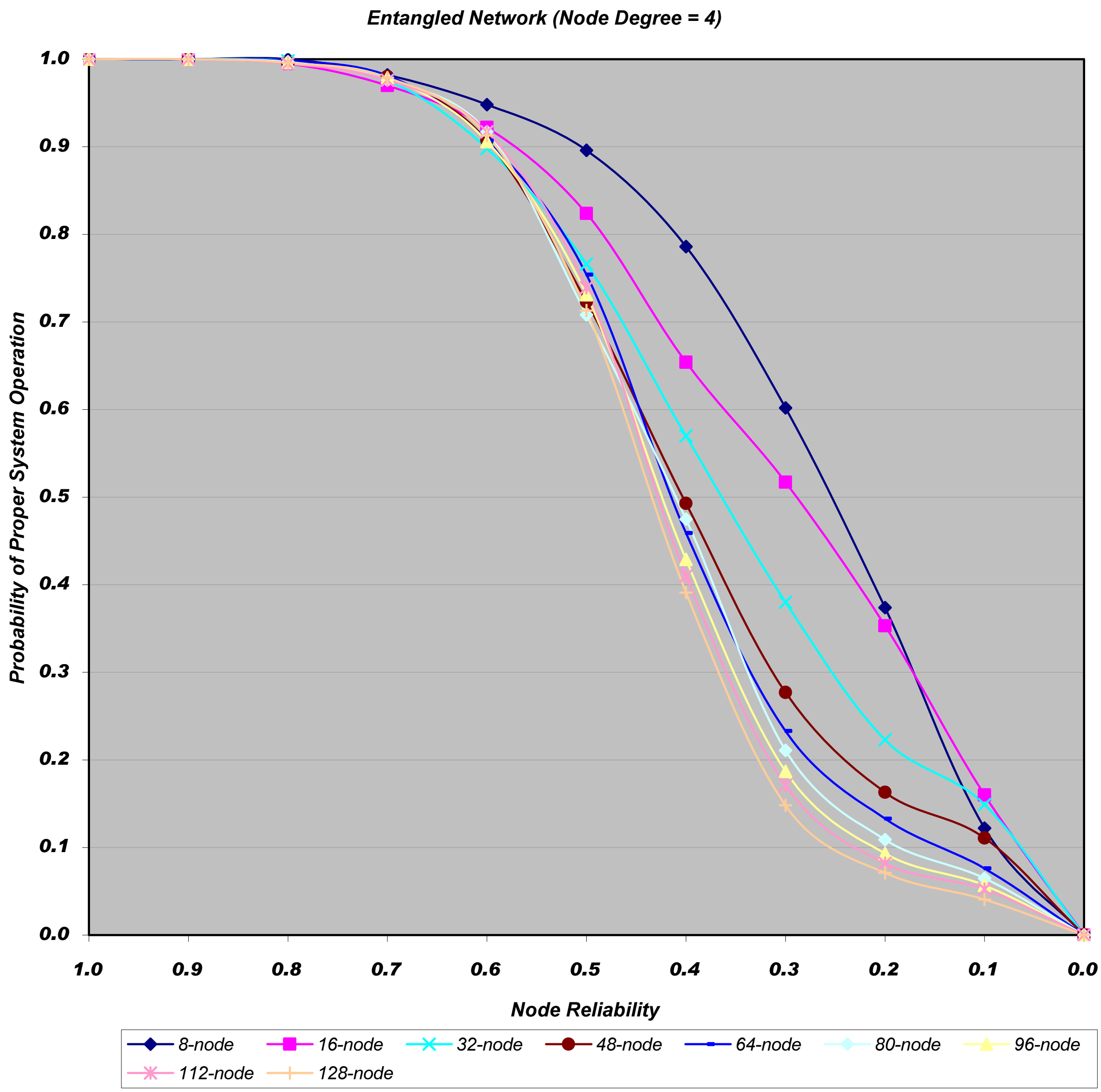

Figure 19. Entangled Network Reliability, Node Degree of 4

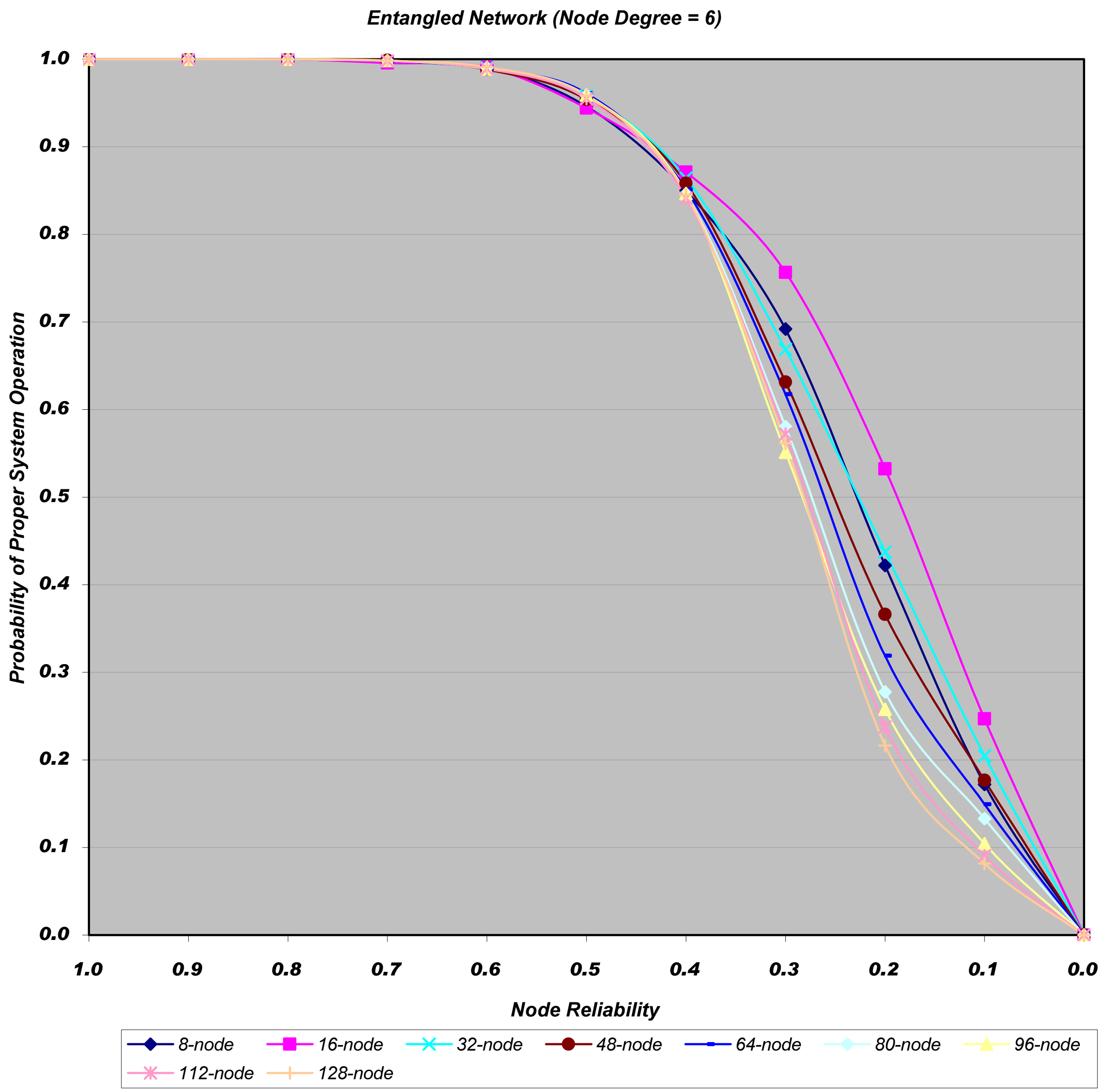

Figure 20. Entangled Network Reliability, Node Degree of 6

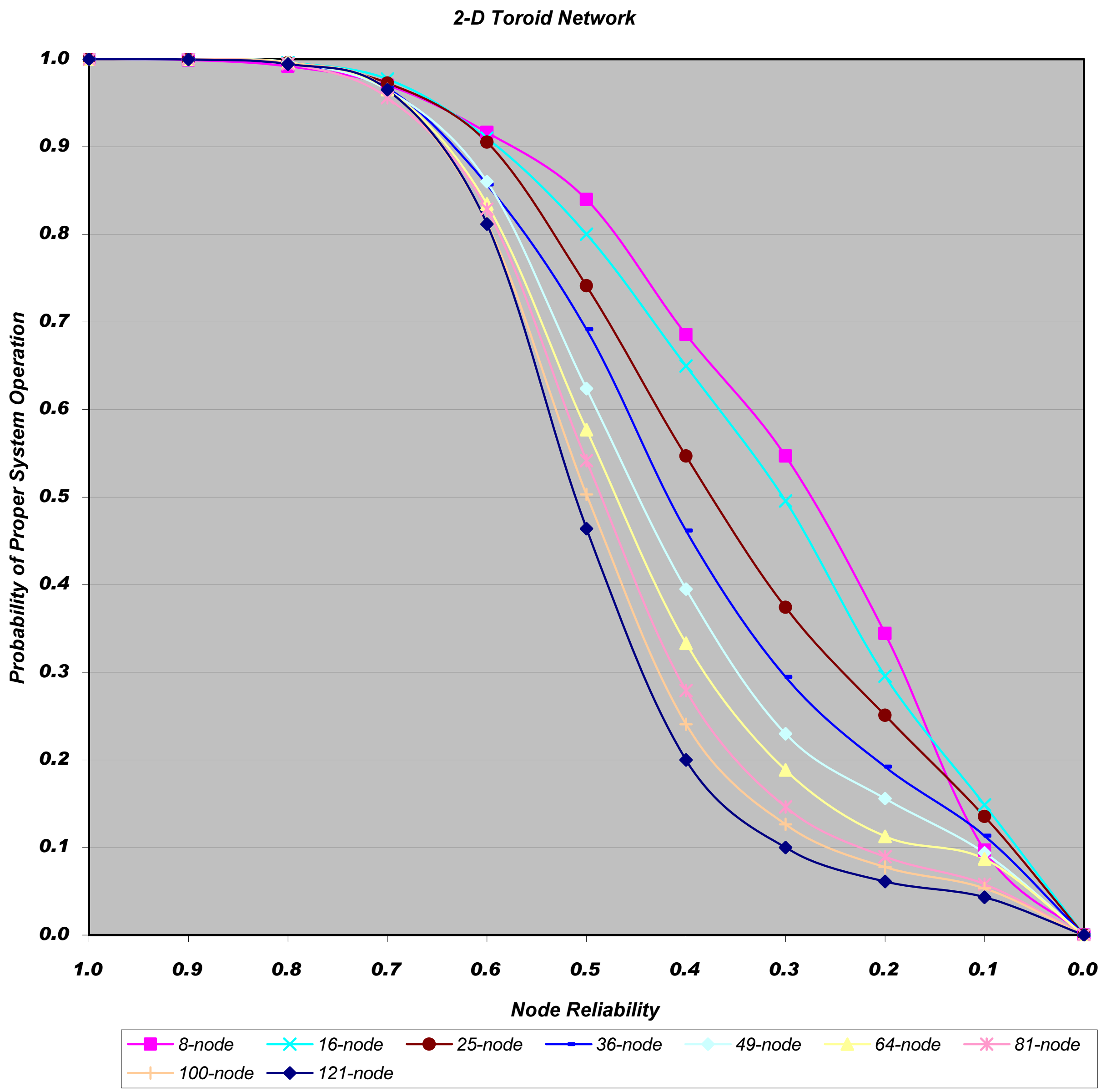

Figure 21. 2-D Toroid Network Reliability, Node Degree of 4

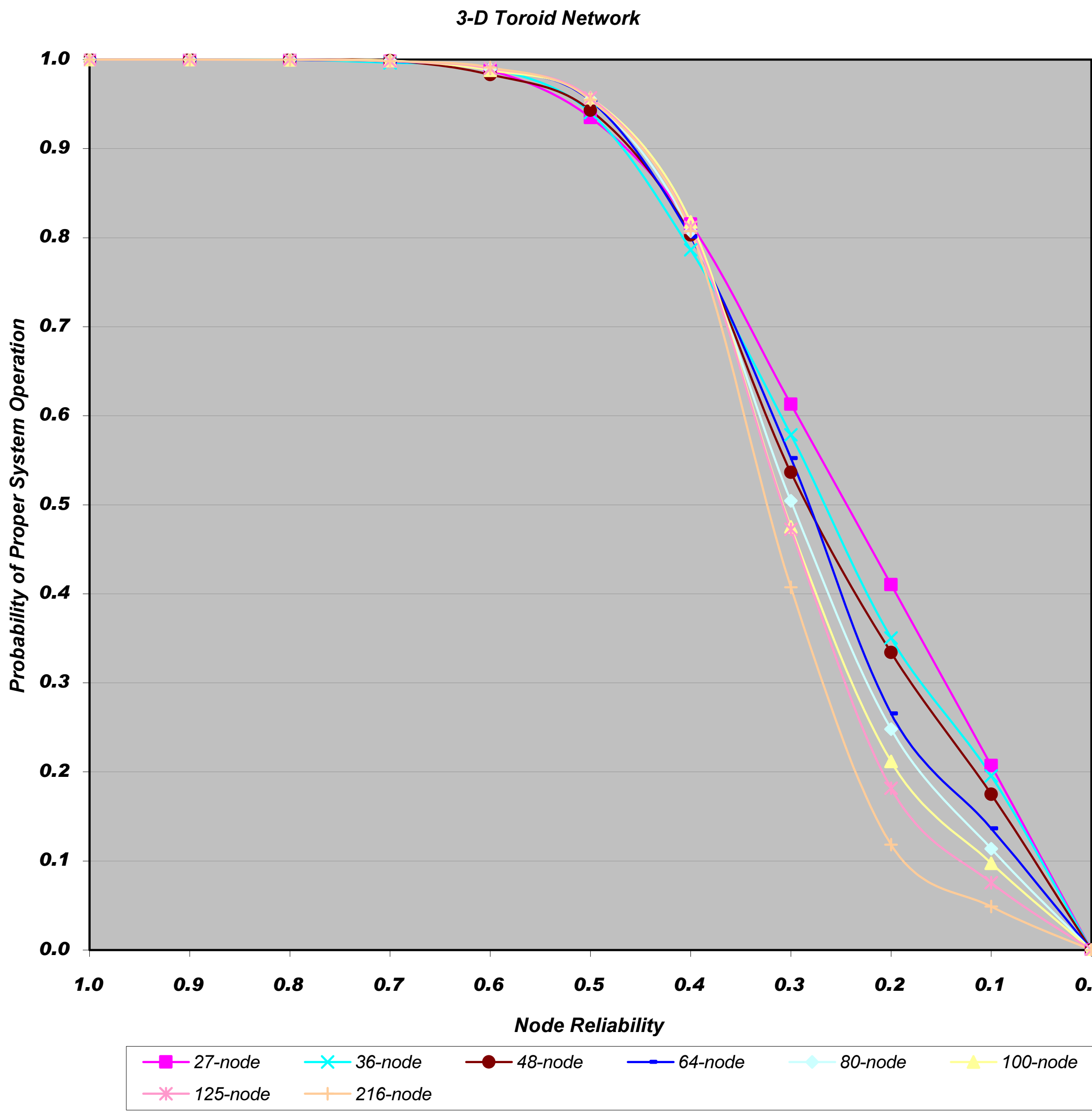

Figure 22. 3-D Toroid Network Reliability, Node Degree of 6

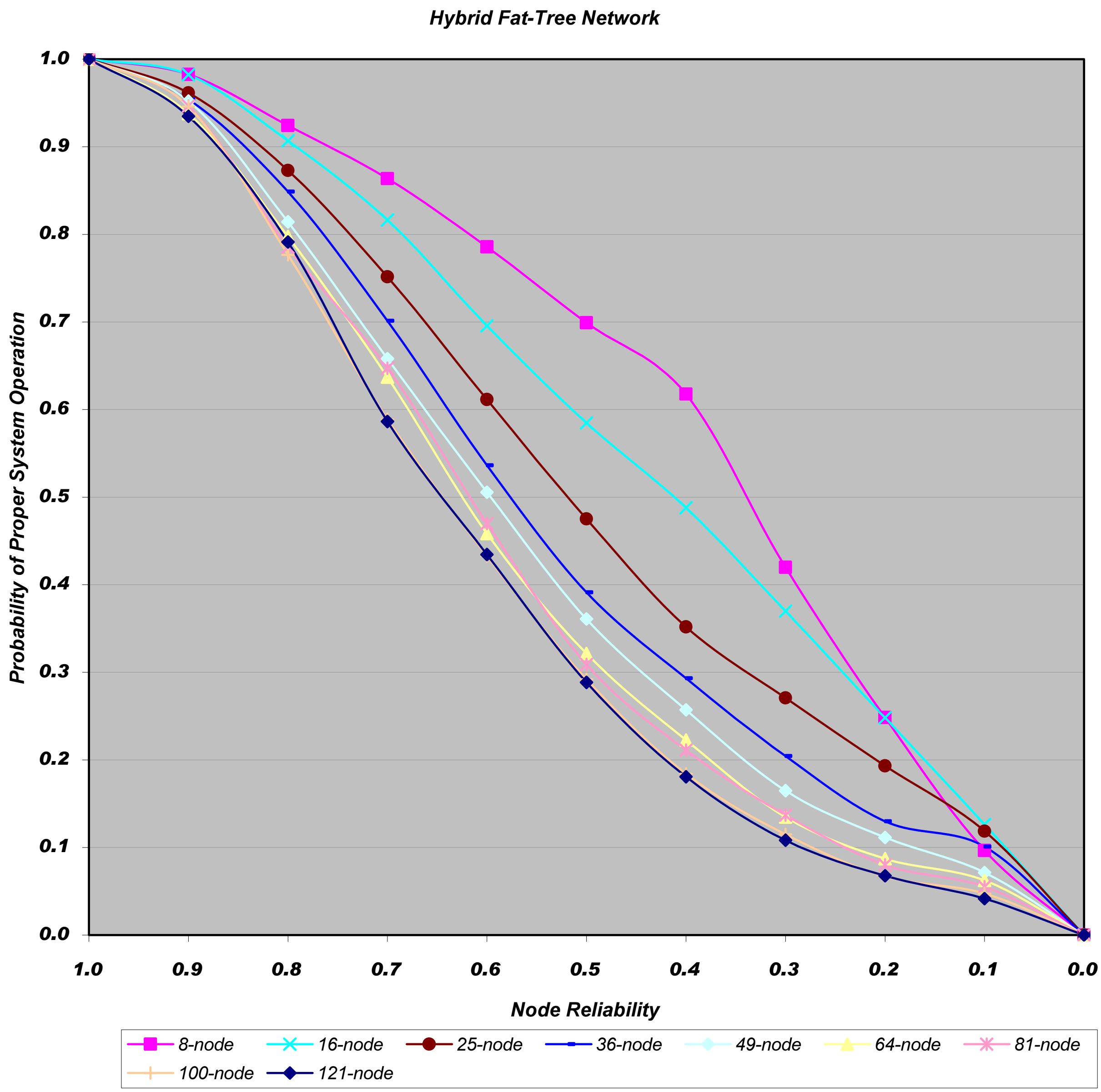

Figure 23. Hybrid Fat-Tree Network Reliability

Figure 24 shows the effect of topology on system reliability. All of these networks have 16 nodes (except for the 3-D toroid, whose minimum size is 27). As you can see, results are loosely grouped according to a topology's node degree. Entangled networks are generally the best performers.

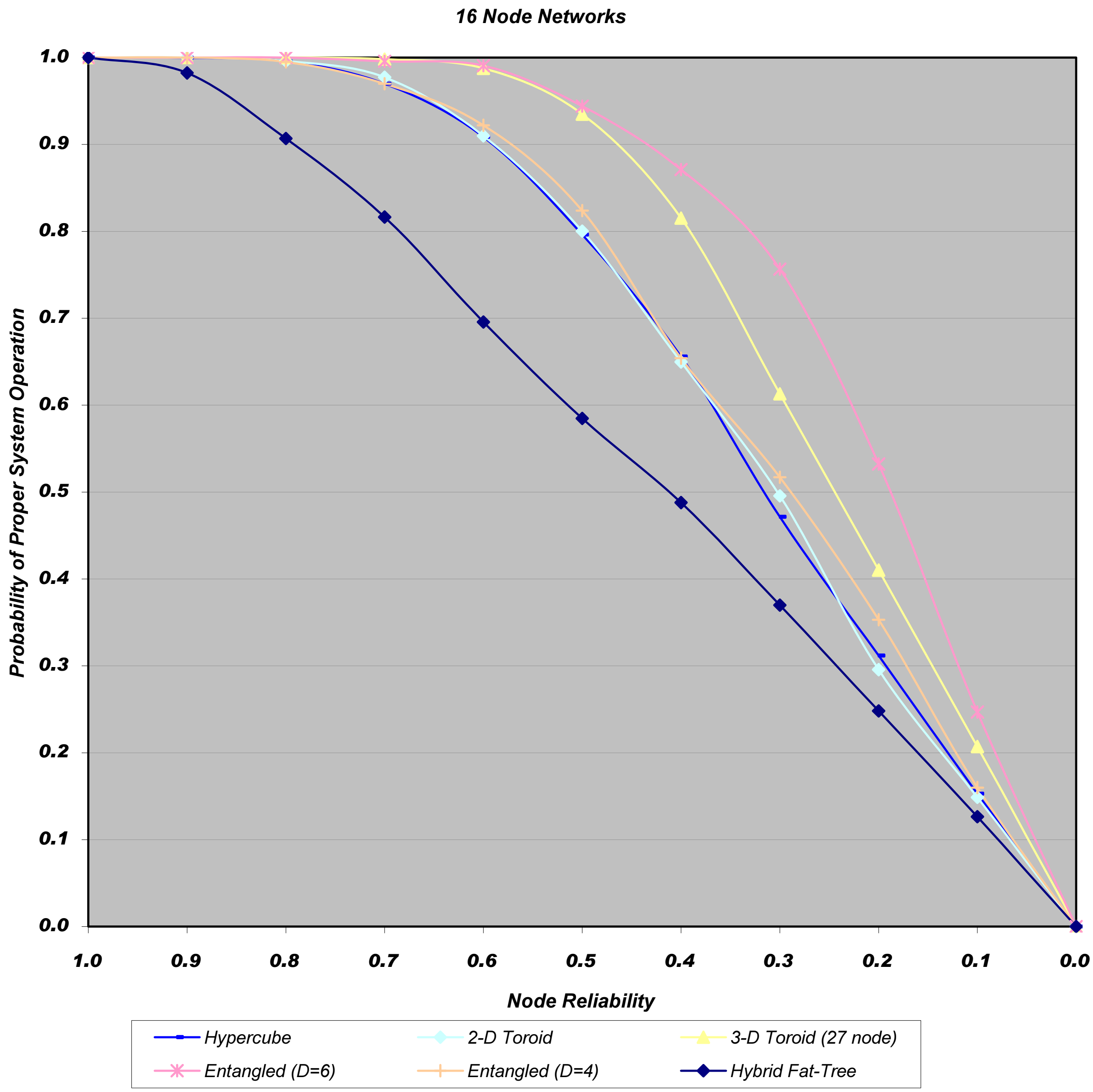

Figure 24. Reliability of Various 16-Node Networks

Figure 25 shows results of the 32-node networks. This graph better demonstrates the relation between node degree and potential reliability. The higher degree topologies (6-way entangled and 3-D toroid) lead the way, followed by the 5-way hypercube, followed by the 4-way entangled and 2-D toroid networks. Remember that topologies with higher node degree merely have more potential for higher reliability – ultimately it is the layout of the nodes in the topology that determines tolerance to failure.

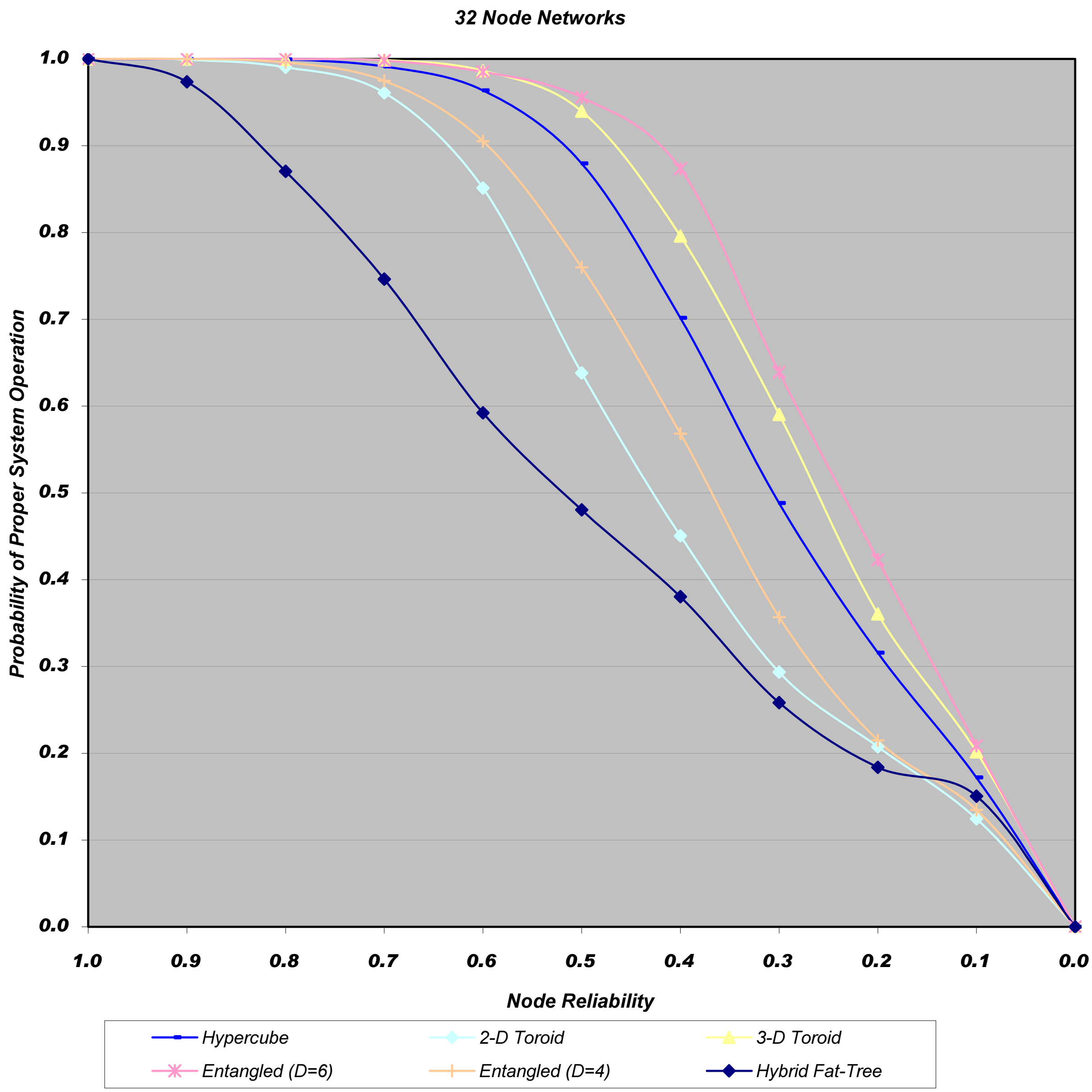

Figure 25. Reliability of Various 32-Node Networks

Figure 26 shows the results of the 36-node networks. Note that the hypercube in this case is an incomplete hypercube. This causes the reliability of the 36-node hypercube model to fall slightly below expected values. The 32-node complete hypercube is also shown on this graph for reference purposes.

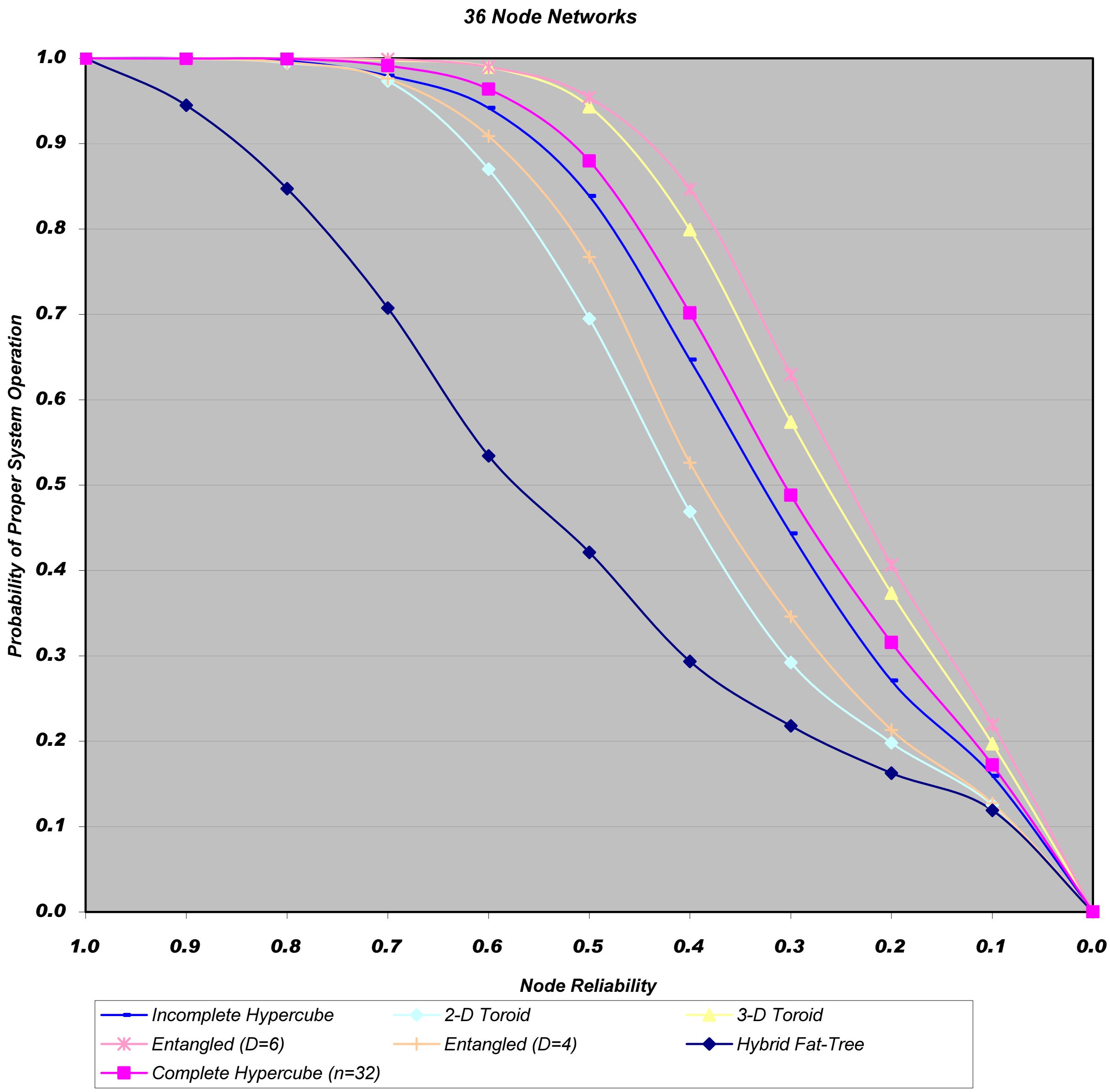

Figure 26. Reliability of Various 36-Node Networks

In Figure 27 the results are observed on 64-node networks. Notice that the hypercube, 3-D toroid, and 6-way entangled network all have equal node degree of 6, and for the most part their results are somewhat similar. Again, though, the entangled networks have a slight edge over their counterparts with equal node degree.

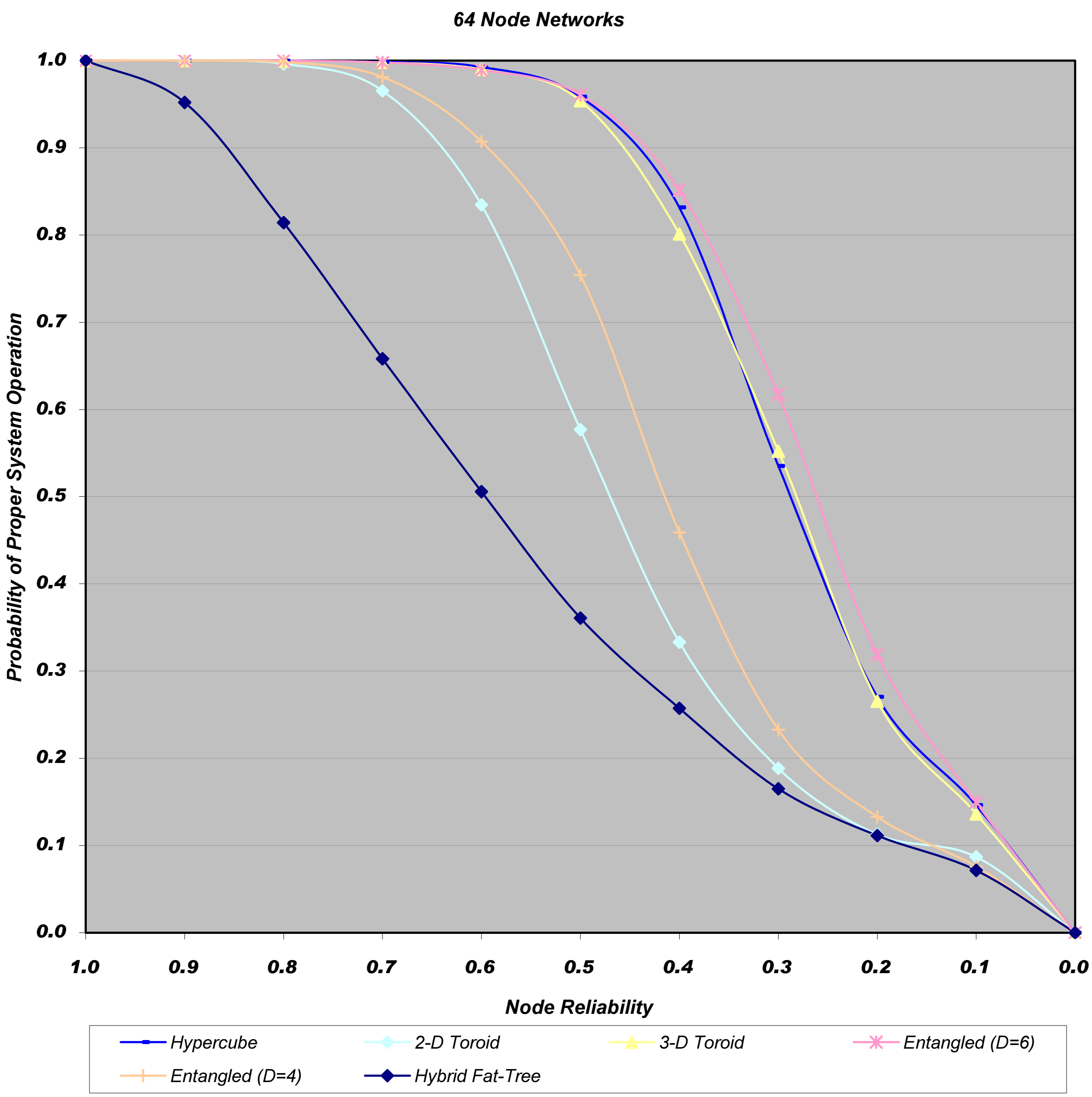

Figure 27. Reliability of Various 64-Node Networks

In Figure 28 the results are observed on 128-node networks. The hypercube, now with a regular node degree of 7, becomes the most reliable topology due to its increased link count and node degree. Not surprisingly, entangled networks still hold the lead over competing topologies with the same node degree.

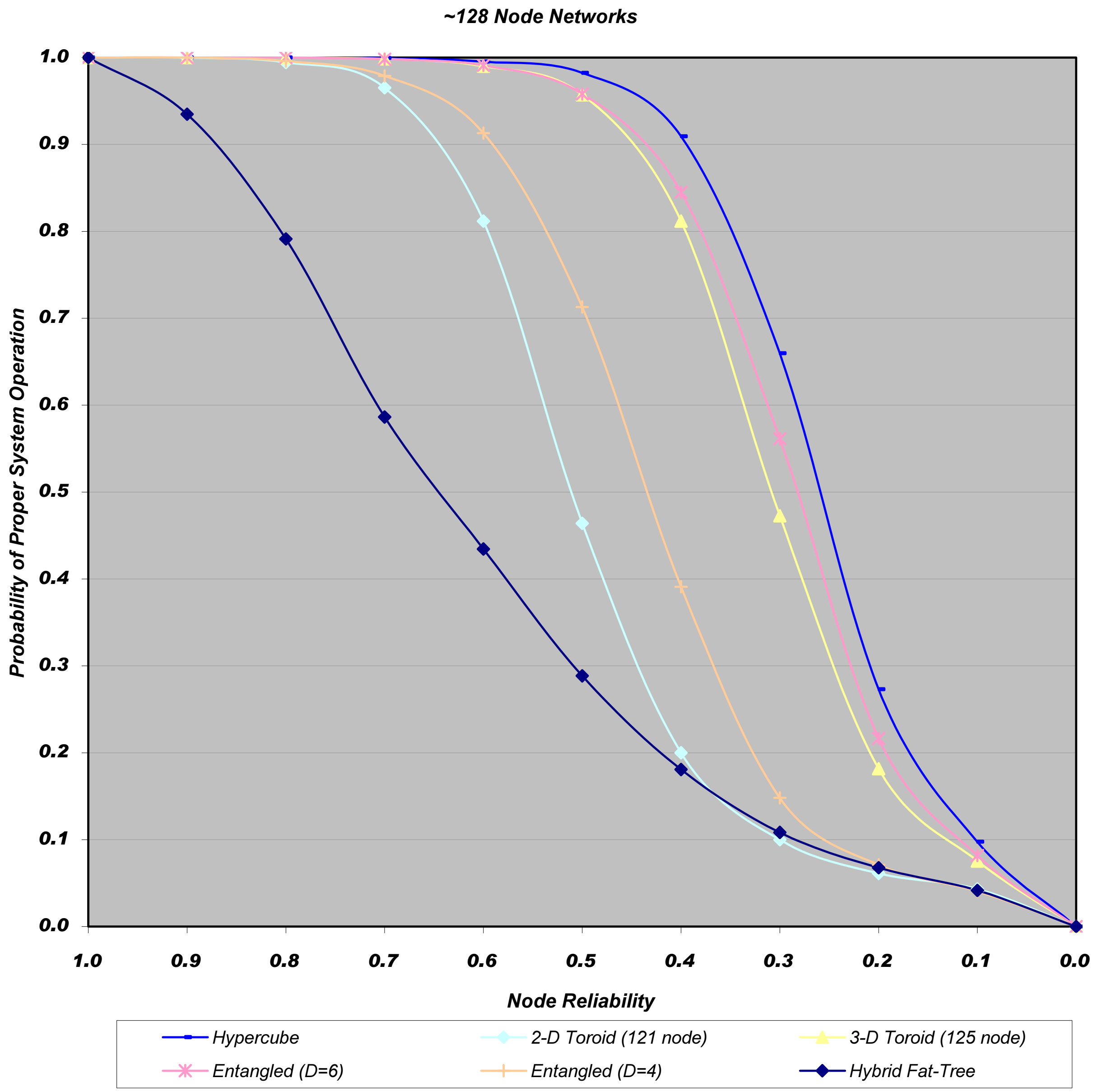

Figure 28. Reliability of Various 128-Node Networks

8 CONCLUSION

The best topologies found during the course of this study were (ordered most desirable to least desirable): Entangled networks, 3-D toroids, complete and incomplete hypercubes, 2-D toroids, cube-connected cycles (for low port counts), and finally fully connected (for low node counts).

Entangled networks performed consistently better than all other topologies of similar node count and degree. These networks are the most promising in terms of reliability, performance, and cost. Arbitrary node counts and arbitrary, fixed node degree make entangled networks very promising for networks of all sizes.

3-D toroids performed well overall. Their structure is both simple and well-defined, and they perform well and reliably. Node degree of 6 is fixed, but may be high for some networks. Network sizes may be somewhat limited by the node count – 3-D toroids require some number of nodes equal to $X*Y*Z$ where $X, Y, Z \geq 3$. This is especially problematic when the number of nodes in the system is equal to a prime number, and cannot be reduced into three non-fractional divisors.

Complete hypercubes are a good solution for medium-node-count networks. Again, there is some limitation on node count (must be a power of 2), but when a network can fit into this topology the performance and reliability are very acceptable. Past ~64 nodes, node degree starts becoming excessively large (>6). For networks that need node counts not equal to a power of two, incomplete hypercubes can offer similar performance and sometimes similar reliability as complete hypercubes. The reliability of incomplete hypercubes, however, does vary with node count (see the graph in Section 4 regarding worst-case connectivity). Therefore, it is best to examine a proposed incomplete hypercube topology before implementing it to ensure that it will meet reliability requirements. Node count values that are slightly below powers of 2 are generally safe choices.

2-D toroids are decent performers with a fixed node degree of 4. There are perhaps better topologies for the same node count and degree, but the simplicity of this topology's layout and routing keep it as a contender for potential implementation.

Cube-connected cycles are perfect when node degrees need to be low (=3) and fixed. When compared to networks of higher node degrees, these networks do not perform well – but for networks with smaller node degrees, these networks excel. In fact, the optimization procedure for entangled networks will sometimes yield a topology equivalent to a cube-connected cycle topology (for some particular node counts when node degree = 3)! The number of nodes required to form this topology is very specific, though, and is a prohibitive factor.

Finally, fully connected networks give the best possible performance and reliability of any topology, but at the cost of very high link counts. This topology is feasible for networks with low node counts, but overall the number of links and the resources required to support those links make this topology not feasible for networks beyond five or six nodes.

9 REFERENCES

[1] V. K. Balakrishnan, Graph Theory. New York: McGraw-Hill, 1997.

[2] A-H. Esfahanian, On the evolution of graph connectivity algorithms, *Selected Topics in Graph Theory*, Edited by Robin Wilson and Lowell Beineke, Cambridge University Press.

[3] A. Ghosh and S. Boyd, Growing well-connected graphs, in *IEEE Conference on Decision and Control*, December 2006.

[4] L. Donetti, F. Neri, and M. Munoz, Optimal network topologies: expanders, cages, Ramanujan graphs, entangled networks and all that, in *Journal of Statistical Mechanics: Theory and Experiment*, August 2006.

[5] Katseff, H. P. Incomplete Hypercubes, Correspondence in *IEEE Transactions on Computers*, Vol. 37, No. 5, May 1988.

[6] A. Harwood and H. Shen, A low cost hybrid fat-tree interconnection network, in *Proceedings of International Conference on Parallel and Distributed Processing and Applications*, 1998.

[7] R. Diestel, *Graph Theory*, 3rd ed. Springer, Germany: Springer, 2005.

[8] Walker, A. and Abachi, H. Reliability Analysis of Tree, Torus, and Hypercube Message Passing Architectures, in *Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory, 1997,* pp. 44-48, March 9-11, 1997.

DISTRIBUTION

1	Leonard G. Burczyk Mail Stop D440 Los Alamos National Laboratory Los Alamos, NM 87545	
4	Office of Nonproliferation Research & Development NA-22/GH-068 Attn: Vaughn Standley W. Randy Bell Roger Byrd Philip Cole 1000 Independence Ave. SW Washington, DC 20585-0420	
1	MS0343	Jim B. Woodard, 2600
1	MS0406	Toby O. Townsend, 5713
1	MS0501	Steve B. Rohde, 5337
1	MS0503	Mythi M. To, 5337
1	MS0530	Dan E. Gallegos, 2623
1	MS0530	Eric Olilla, 2623
1	MS0801	Dorothy S. Rarick, 9330
1	MS0806	John M. Eldridge, 9336
1	MS0806	Len Stans, 9336
1	MS0806	Jason S. Wertz, 9336
1	MS0971	Bob M. Huelskamp, 5730
1	MS0971	Jae W. Lee, 5733
1	MS0972	Kurt R. Lanes, 5560
1	MS0986	J. (Heidi) Ruffner, 2664
1	MS0964	Brian C. Brock, 5733
1	MS0980	Matt P. Napier, 5571
1	MS0980	Steve M. Gentry, 5703
1	MS0982	Dan E. Carroll, 5732
1	MS0982	Dan Kral, 5732
1	MS0982	Phil J. Cole, 5732
1	MS0986	Dave M. Bullington, 2664
1	MS0986	Jonathon W. Donaldson, 2664
1	MS0986	John V. Vonderheide, 2660
1	MS0986	Jeff L. Kalb, 2664
1	MS0986	Dave Heine, 2664
1	MS0986	Dave S. Lee, 2664
1	MS1003	Ray H. Byrne, 6473
1	MS1172	Ron J. Franco, 5415
1	MS0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories