Breeding Diameter-Optimal Topologies for Distributed Indexes

Sanket Patil
Srinath Srinivasa
Saikat Mukherjee
Aditya Ramana Rachakonda
Open Systems Laboratory
International Institute of Information Technology
Bangalore - 560100, India

Venkat Venkatasubramanian
Laboratory for Intelligent Process Systems
School of Chemical Engineering
Purdue University
West Lafayette, IN 47906, USA

The role of a distributed index from the perspective of an individual actor (node) is to minimize its separation from all other actors (nodes). From a systemwide perspective, an optimal distributed index is one that minimizes the diameter of the index graph. We tackle this optimization problem in an evolutionary fashion by performing a series of topology crossovers and fitness-based selections. A set of constraints regulate the fitness function. Different classes of topologies such as star, circle, and skip lists emerge as diameter-optimal structures under different constraints. Knowledge of the optimal topology class in a given context provides strategic information for distributed agents to (re)construct a global index structure based on local information. We also investigate a deterministic approach called *polygon embedding*, to build topologies with similar properties to that of the evolved topologies.

1. Introduction

Increasingly, large information systems are modeled as systems of autonomous agents that are distributed over wide geographical areas. The agents work independently of one another. They are each aware of a small number of other agents with whom they collaborate and/or compete. In addition, there are also likely to be several partially connected agents such as mobile hosts or dial-up connections that are peripherally connected with the system.

A crucial element of performance in such systems is the *distributed index*. This is an index structure that is spread across the system and used for data lookup or routing information. For purposes of load balancing and fault tolerance, such index structures are distributed across all machines in the system (or at least across all the reliable hosts in the system). The efficiency of the index, and that of the

https://doi.org/10.25088/ComplexSystems.18.2.175

system itself, is determined by the *lookup* complexity. This is a bound on the number of lookups or network connections that an application program needs to make, before it finds the required data element.

Unlike conventional index structures that are based on variants of search trees, distributed index structures do not have a single point of entry. Any tree-like distributed data structure would mean that the machine hosting the root node would have to bear a disproportionately large proportion of the lookup load.

Distributed indexes are hence typically designed as graphs where lookup requests can start from anywhere in the data structure. The optimization objective is to limit the number of lookups from anywhere to anywhere else in the distributed system.

The design is also constrained by several other factors. The number of links that constitute the index poses a cost on bookkeeping and network connections. The skew in the distribution of links poses issues of load balancing. The lack of reliability on the part of some machines poses issues of resilience of the index in the face of failures.

One way of addressing this problem is to provide each agent in the system with a set of edges and ask it to autonomously make connections with other agents. These connections would be based on maximizing an individual self-interest function, such as minimizing the agent's separation with all other agents in the system. The connections would be constrained by factors such as maximum allowable degree. While this is optimal locally, the resultant topology may not be optimal globally. From a systemwide perspective, the optimization objective is to minimize the diameter of the index graph.

Given an arbitrary set of nodes and constraints, to the best of our knowledge, there is no deterministic procedure for obtaining a diameter-optimal topology. Thus, one of the goals of this work is to find useful deterministic or heuristics-based procedures to obtain diameter-optimal topologies.

The topology design problem has received significant interest in the area of data-centric peer-to-peer networks. Several distributed hash tables (DHTs) such as Chord [1], Pastry [2], and Koorde [3] use distributed index structures to look up where a given hash bucket is located. Distributed indexes continue to be an area of active research [4-7].

In the context of peer-to-peer networks, a high emphasis is placed on the "symmetric" nature of the data structure. All peers are assumed to have nearly equal degree in forming the DHT. Given this requirement on a network of n nodes, a lookup complexity of $O\left(\frac{\log n}{\log\log n}\right)$ is required for a network having high fault tolerance [3]. This entails a uniform degree distribution of $O(\log n)$ per node.

When a symmetric design is not of importance, scale-free networks with power-law degree distributions have been shown to be optimal in the sense of balancing conflicting goals such as minimizing network diameter, minimizing infrastructure cost, and maximizing robustness

https://doi.org/10.25088/ComplexSystems.18.2.175

[8]. Also, for several decentralized systems such as data grids, it makes more sense to minimize diameter by sacrificing uniform degree distribution. In these scenarios, some of the nodes act as "special peers" or "super nodes" and have a relatively larger load than other nodes in the grid.

However, given an arbitrary set of constraints on cost and load distribution, obtaining the optimal topology is an intractable problem. To address this problem, we look at topology design as an evolutionary optimization problem. The objective is diameter minimization subject to infrastructure cost, bookkeeping cost, and load distribution constraints.

We employ genetic algorithm techniques to perform the evolutionary optimization. Several initial populations are considered where actors make connections with other actors in order to minimize their separation from the rest of the network. A selection function chooses topologies based on their global fitness that is in turn a function of the graph diameter. The evolutionary process proceeds by crossing over topologies from the current population to obtain the next set of offspring topologies. The process stops when the topologies converge.

The outcomes of the breeding process show different families of diameter-optimal topologies that arise due to changes in the three parameters: infrastructure cost, bookkeeping cost, and load-distribution requirements.

For a network of n nodes, when the maximum number of edges is limited to $n - 1$ or the *infrastructure cost* k is minimal, we get a family of trees, which converge to the star topology when the maximum permissible degree of nodes, or the bookkeeping cost p, is unbounded and a high emphasis is placed on efficiency η. A circular topology emerges when the number of edges is restricted to n and load distribution ρ is given high importance. We obtain a family of circular skip lists when the maximum number of permissible edges in the network is greater than n. The maximum number of edges in the network cannot be less than $n - 1$ as it would disconnect the graph. As load distribution is given more prominence, the scale-free topology gives way to a symmetric topology (or a regular graph topology), with uniform degree distribution.

Genetic algorithm optimization is a randomized process and it is difficult to discern patterns of connectivity in the emergent topologies. As a result, it is difficult to design algorithms based on local knowledge, such as navigation rules. In order to address these problems, we propose a second strategy called *polygon embedding*, which is the embedding of a polygon of a predetermined size in a network formed by arranging all the nodes in a circle. In other words, it is a circular skip list with the polygon edges forming the chords. For every topology that is bred, we determine polygon embeddings that have similar characteristics for efficiency, load distribution, and cost.

We assume that edges are undirected. We work with undirected edges because in application domains such as data-centric networks,

https://doi.org/10.25088/ComplexSystems.18.2.175

grids, and so on the cost of making an edge undirected (or bidirected) is not significant. Certain optimal topologies such as DHTs generally consider directed graph topologies. Therefore, in order to be consistent with them, when comparing our topologies with existing DHT topologies, all undirected edges can be considered as two oppositely directed edges, and the infrastructure cost can be doubled.

2. Related Literature

Diameter or lookup optimality has been addressed by several DHTs. In Chord [1], an identifier space is formed as a logical circle, to which both nodes and keys are mapped using a hash function. The Chord topology is a skip list, wherein each node connects to $\log n$ other nodes on the circle to achieve a diameter of $\log n$ and symmetric degree distribution. When symmetry in degree distribution is not important, we can achieve a $\log n$ diameter at much less cost as shown later in Section 5. Koorde [3] describes an implementation of a De Bruijn graph-based network, in which a $\log n$ diameter is achieved with a fixed degree of 2. That paper also describes an extension to a degree-m De Bruijn graph; by using this extension, a diameter of $\log_m n$ can be achieved. D2B [4] is another implementation of the De Bruijn graph that is similar to Koorde.

There are different kinds of regular graphs described in the literature that find applications. In a connected m-regular graph, each node has degree m. Regular graphs are interesting to us because of their symmetric load distribution. In our experiments, when load distribution is given high importance, topologies tend toward regular graphs. A hypercube graph is a regular graph of 2^m nodes, represented by all m-length binary strings. Each node connects to all other nodes that are at a Hamming distance of 1, forming an m-regular graph. A hypercube graph has a diameter of m, which is the maximum Hamming distance between any two nodes.

A De Bruijn graph is a directed graph where each node is mapped onto an identifier in the identifier space formed by all m-length strings of an alphabet of length b. Every node has exactly m outgoing edges. The m edges are drawn by right shifting each node identifier by one position, and adding each of the b symbols in the alphabet at the end. A De Bruijn graph guarantees a diameter of $\log_b(b^m)$. It is a regular graph in the sense that every node has the same indegree and outdegree. However, an undirected version of the De Bruijn graph is diameter-suboptimal, even when undirected edges are considered as single edges.

The Moore bound [9] defines an upper bound on the number of nodes that can be packed in a graph of fixed degree m and diameter d. To estimate this, imagine an m-ary tree of depth d. So, the maxi-

https://doi.org/10.25088/ComplexSystems.18.2.175

mum number of nodes $N_{\max}$ can be estimated as: $N_{\max} = 1 + m + m^2 + m^3 + \cdots + m^d = \frac{(m^{d+1}-1)}{m-1}$.

As a consequence, the lower bound on the diameter of a graph with n nodes and a fixed degree m is estimated as $D_{\min} = \lceil \log_m (n\,(m-1)+1) \rceil - 1$.

A Moore graph is an m-regular graph that has a diameter d, in which the number of nodes is equal to the Moore bound. Though Moore graphs are optimal structures, it is only possible to construct trivial Moore graphs.

Loguinov [5] provides a detailed graph-theoretic analysis of peer-to-peer networks, with respect to routing distances and resilience to faults. That paper argues that De Bruijn graphs offer the optimal diameter topology among the class of practically useful graphs because of their low diameter. Also, they come close to satisfying the Moore bound. Being directed graphs, De Bruijn graphs are well suited for applications like DHTs. However, it can be shown that an undirected De Bruijn graph is not the best possible topology in terms of diameter-optimality. Also, the topology of De Bruijn graphs are fixed and cannot be altered to have better diameters by compromising on the degree distribution. Therefore, De Bruijn graphs cannot be claimed to be optimal in a general sense.

In HyperCuP [10], a hypercube graph is constructed in a distributed manner by assuming that each node in an evolving hypercube takes more than one position in the hypercube. That is, the topology of the next dimensional hypercube implicitly appears in the present hypercube, with some of the nodes also acting as “virtual” nodes to complete the hypercube graph. Similarly, when nodes go away, some of the existing nodes take the missing positions along with their own.

Viceroy [11] is an implementation of an approximate butterfly. Nodes are arranged in $\log n$ levels, with the nodes at each level forming a ring topology with each node having an outgoing link to a successor and a predecessor. Apart from the “neighbors” on the ring, each node has long-range outgoing links to five other nodes across the $\log n$ levels. These levels and nodes are chosen by a randomized process. Viceroy claims to achieve a $\log n$ diameter with a fixed degree of 7. Ulysses [12] is another implementation of the butterfly, though with its $\log n$ neighbors, it is not a fixed degree graph.

A distributed trie-based approach is proposed in [13] that is based on the prefix-based routing of Plaxton et al. [14], wherein a k-ary prefix tree is maintained in a distributed manner. The maximum degree of a node is $k+1$ and the diameter is $2\lceil \log_k n \rceil$. A content addressable network (CAN) [15] forms an identifier space over an approximation of an m-dimensional torus. A CAN has a fixed degree of $2\,m$ and provides a diameter of $\frac{m\,n^{1/m}}{2}$.

https://doi.org/10.25088/ComplexSystems.18.2.175

Pandurangan et al. [16] propose an algorithm to dynamically construct a DHT under a probabilistic model of node arrivals and departures. The DHT has a fixed low degree and achieves a logarithmic diameter almost certainly. However, similar to our topology breeding approach, it does not have a deterministic navigation algorithm. Also, their algorithm is partially centralized, since new nodes register with a central server and the server selects a set of random nodes to connect to, rather than the node doing it. Law et al. [17] take up the problem of the distributed construction of random regular graphs that have m Hamiltonian circuits. The graph is $2m$-regular with each node having a degree of $2m$, and m Hamiltonian circuits passing through each node. With a high probability these graphs will achieve a diameter of $\log_m n$.

Kleinberg talks about a small world [18], where the nodes are arranged in a metric space having a distance function defined across any pair of nodes. Each node is modeled as having short-range links to its neighbors until a certain range, plus some random long-range links to nodes that are far away in the lattice. Kleinberg argues that although we can find several families of such small-world topologies that have short routing paths, it is very difficult to find short paths between pairs of nodes based on local information. He goes on to prove that there is only a unique family of small worlds for which a decentralized navigation algorithm can be constructed. Symphony [19] is inspired by Kleinberg's model. The Symphony topology is a ring with a set of long-range links, which are randomly drawn from a family of continuous harmonic distributions.

Ratnasamy et al. [6] identify the trade-offs between routing table length (degree) and length of lookup (diameter). They argue that DHTs with a $\log n$ degree achieve a $\log n$ diameter, whereas fixed-degree DHTs, say the degree is m, achieve a diameter of $n^{1/m}$. They ask if these are the optimal trade-offs, and if they are not, will topologies that improve this trade-off do so at the cost of some other desirable properties. Xu et al. [7] take off from there. They argue that congestion (load) in the network is an important parameter that will be affected badly by trying to improve the trade-offs from [6]. They conjecture that for a uniform load distribution, the given trade-offs are asymptotically optimal. However, when uniform load distribution is not one of the desired properties, we can easily develop DHTs with better degree and efficiency performance. Gummadi et al. [20] consider "flexibility" in neighbor and route selection that different DHT topologies offer. They conclude that the ring structure performs the best. Again, the circular skip lists that result from our breeding experiments corroborate this finding.

A common feature across the cited literature is that the different optimal topologies have been proposed for some class of graphs keeping some limiting constraint or the other. The proposed work can provide a new perspective to the topology design problem by addressing

https://doi.org/10.25088/ComplexSystems.18.2.175

it in a more general fashion and employing an evolutionary optimization approach. Our focus is not on designing optimal topologies for particular applications, but to develop a general understanding of the problem.

This work is similar in spirit to the work of Venkatasubramanian et al. on the spontaneous emergence of optimal networks [8]. In their work, a network of nodes is allowed to evolve, over several generations, to form a "fit topology" that satisfies certain survival objectives. The "fitness" of a network topology is defined as a function of efficiency, robustness, and cost. By varying these parameters, they get a set of topologies that are optimal in different scenarios. Further, they use the knowledge of these topologies to develop a theory of complex teleological systems—both biological and human engineered [21].

Taking the work of Venkatasubramanian et al. as our point of departure, we study different families of "fit" topologies. For our purposes we define *efficiency* as the diameter of the resulting graph, *load distribution* is defined in terms of the skew in degree distribution across the nodes, and *cost* is defined as the number of edges in the graph. We also define a maximum degree constraint on nodes. By varying the maximum degree and the number of edges we allow the network to optimize on diameter and load distribution.

3. Breeding Optimal-Diameter Topologies

In this section we formally define our optimization parameters and then describe the topology evolution technique.

3.1 Definitions

Efficiency (η): Efficiency is a measure of how good the diameter of a topology is. The worst diameter for a connected graph of n nodes is $n-1$, which is the diameter of a straight line graph, and the best is 1, which is the diameter of a clique. In other words, a topology is most efficient if the diameter is 1, and least efficient if it is $n-1$. We map a diameter d that falls in the interval $[1, n-1]$, to a value of efficiency in the interval $[0, 1]$, as:

$$\eta = 1 - \frac{d-1}{n-2}.$$

By this definition, a straight line topology has an η of 0, a circular topology has an η nearing 0.5, a clique has an η of 1, and so on.

Load distribution (ρ): Load distribution is measured in terms of the skew in degree distribution. We define this as the difference in the maximum degree in the graph $\hat{p}$ and the mean degree of the nodes $\overline{p}$.

https://doi.org/10.25088/ComplexSystems.18.2.175

For a connected graph of n nodes, the worst skew occurs for the star topology. The central node has a degree of $n-1$ and all the nodes surrounding it have a degree of 1. Therefore, the worst skew is $\frac{(n-1)(n-2)}{n}$. The best skew is 0, when all the nodes have the same degree. This occurs when the topologies are regular graph topologies as in a circular topology or a clique.

Thus, load distribution is a mapping from a value in the interval $\left[0, \frac{(n-1)(n-2)}{n}\right]$ to a value in the interval [0, 1]:

$$\rho = 1 - \frac{n\left(\hat{p} - \overline{p}\right)}{(n-1)(n-2)}.$$

By this definition, a star topology has a ρ value of 0 and regular graph topologies have a ρ value of 1.

Maximum permissible degree (p): The maximum permissible degree p is an upper limit on the number of edges that can be incident on a node. It is a measure of the local "bookkeeping cost" (e.g., the size of the local finger table at a DHT node).

Infrastructure cost (k): Infrastructure cost is defined as a function of the number of edges e in the graph. The minimum number of edges $e_{\min}$ required to have a connected graph is $n-1$. We do not associate any cost to a minimally connected graph. Any "extra" edge has an associated cost. All extra edges cost the same. A clique has the highest cost, with $\hat{e} = \frac{n(n-1)}{2}$ number of edges. Thus, the cost of a topology is defined as a ratio of the number of extra edges to the number of extra edges in the clique with the same number of nodes:

$$k = \frac{e - e_{\min}}{\hat{e} - e_{\min}}.$$

3.2 Fitness

The fitness of a graph is defined in terms of the optimization dimensions: efficiency, load distribution, and cost:

$$\phi = \alpha\rho + (1-\alpha)\eta - k.$$

Here, $0 \leq \alpha \leq 1$, is an application-dependent parameter that acts as a slider between efficiency and load distribution. A high α value indicates that a high emphasis should be placed on the load distribution of topologies during the breeding process.

https://doi.org/10.25088/ComplexSystems.18.2.175

3.3 Breeding

We start with a set of random seed graphs and let them evolve over several generations using crossovers and mutations. Eventually, we choose the fittest topology, as per the fitness function.

We start with a large number of seed graphs (typically 10s of n). The seed graphs are generated randomly under the p and k constraints. We retain only the connected seed graphs. Next, we do an "all-pairs crossover" among these "parents", to form the first generation of "offsprings". Random mutations involving realignment of edges is performed so that the p and k constraints are not breached. Again, after retaining the connected offsprings, we sort them based on their fitness. The next generation of parents is formed by an $80:20$ heuristic, wherein 80% of the new population is formed by choosing the fittest offsprings, and 20% by randomly choosing from the set of lesser fit offsprings. Mixing some less fit offsprings in the population is an effort toward reducing the chances of local minima. The whole process is repeated over several generations.

Figure 1 shows a sample of optimal topologies that we obtained under different constraints. The first row shows various tree structures. Trees are the most optimal topologies when the constraint on infrastructure cost is $k = 0$ ($e = e_{\text{min}}$). It can be seen that the diameter (and hence η) improves along the p axis, until it becomes 2 for a star topology. However, load distribution ρ reduces at the same time. Generally, we are not interested in tree structures since they essentially offer "extreme" properties. The most efficient topology is the least robust and vice versa.

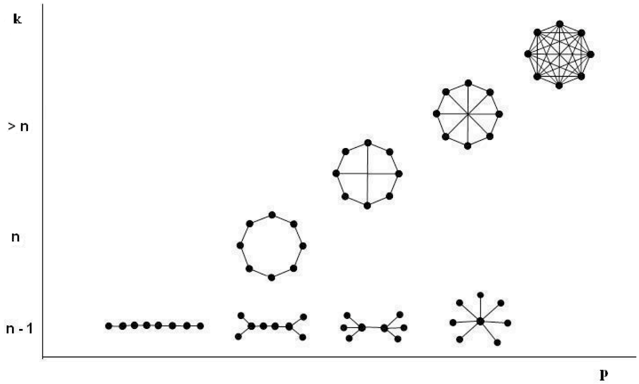


Figure 1. Optimal topologies under different constraints.

When $k \geq n$, which is pertinent for most practical applications, we get what are called "circular skip lists" or simply "skip lists". A skip list is a topology in which each node is given one or more edges and it

https://doi.org/10.25088/ComplexSystems.18.2.175

uses them to connect to nodes at different distances (or skips) on a logical circle. In graph-theoretic terms, the graph can be organized as a Hamiltonian circuit with several chords.

We can see different skip lists in Figure 1. The circle is the minimal skip list, where each node is given a single (undirected) edge, and it connects to a node at skip 1 (a neighbor on the circle). As we start relaxing the constraints on k and/or p, we get a variety of skip lists, until the clique is formed. The clique has the best η and ρ, however at an infrastructure cost of $k = \frac{2n}{n-1}$ and a maximum permissible degree of $p = n - 1$. The figure shows topologies for a very small network ($n = 8$), which is meant merely to serve as an illustration of skip lists. A detailed analysis is presented next.

4. Analysis

Figure 2 shows how efficiency η changes when the permissible bookkeeping cost p is increased. Here, the infrastructure cost k is set to $n - 1$ resulting in a family of tree structures. Also, $\alpha = 0$ indicating that the genetic algorithm places maximum emphasis on efficiency and none on load distribution. The highest value of efficiency for each curve corresponds to the star topology.

Figure 3 shows how load distribution changes with increasing p for $k = n - 1$. Here too, α is set to 0. As is apparent, load distribution goes down linearly as the tree structure approaches a star.

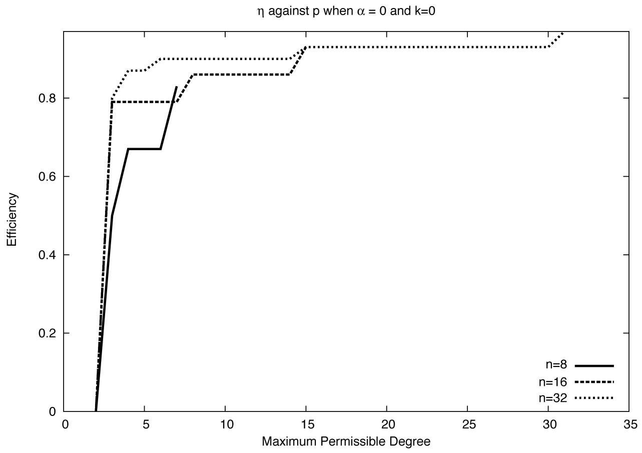


Figure 2. Efficiency against maximum permissible degree for different n when $\alpha = 0$.

https://doi.org/10.25088/ComplexSystems.18.2.175

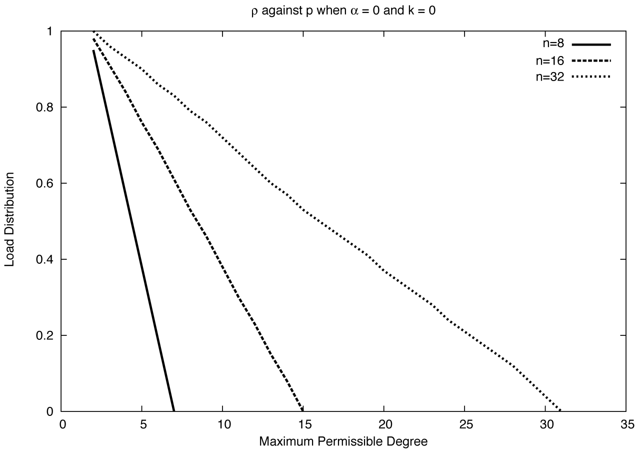


Figure 3. Load distribution against maximum permissible degree for different n when $\alpha = 0$.

In Figure 2, η starts at 0, for $p = 2$ and $e = n - 1$ (corresponding to a straight line), and increases with p until it reaches its maximum value at $p = n - 1$ (corresponding to a star). Similarly, Figure 3 shows the fall in ρ under the same constraints. The value of ρ is high initially despite α being 0 as an effect of a low p. But with an increasing p, ρ goes down to reach 0, when the topology settles at a star.

Figure 4 shows changes in efficiency and load distribution when the maximum number of edges is increased. Here n was set to 32 and $p = 4$. α was set to 1, giving maximum importance to load distribution. We can see that the load distribution values remains close to 1, which is natural since $\alpha = 1$. The efficiency curve, on the other hand, slowly rises as a side effect of the increase in the number of edges, though there is no emphasis on increasing the efficiency.

The most important observation from the topology breeding experiments is that, diameter-optimal topologies that balance cost and load distribution constraints seem to contain at least one Hamiltonian circuit. If we rearrange the topology with one of the Hamiltonian circuits forming the outer circle, we can see circular skip lists.

Figure 5 shows a sample of optimal skip list topologies that evolved during our experiments.

Several existing optimal DHT topologies such as the ring (Chord [1]), butterfly (Viceroy [11], Ulysses [12]), and hypercube (HyperCuP [10]) can be seen as variants of circular skip lists. Our results not only corroborate the DHTs, but also help us make a stronger claim: in general, an optimal communication topology has one or more Hamiltonian circuits.

https://doi.org/10.25088/ComplexSystems.18.2.175

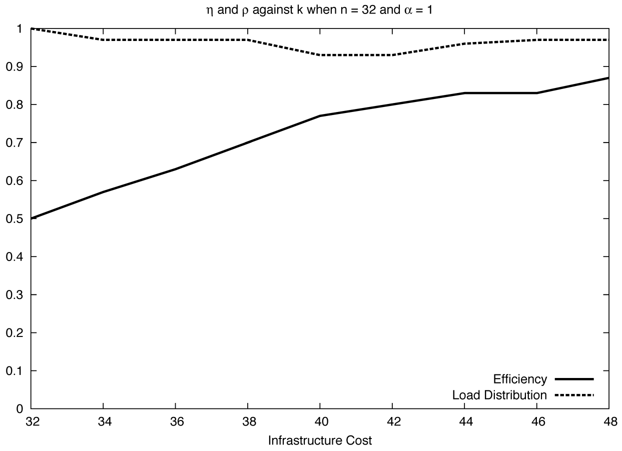


Figure 4. Efficiency and load distribution against infrastructure cost for $n = 32$ when $\alpha = 1$.

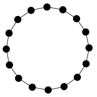
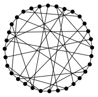
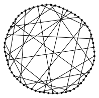
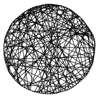
Topology	n	α	η	ρ	ϕ
	16	1	0.5	1.0	1.0
	32	1	0.9	1.0	0.95
	64	1	0.9	1.0	0.99
	128	1	0.97	1.0	0.98

Figure 5. Example optimal topologies that have been bred.

5. Finding Polygon Embeddings with Similar Properties

The skip list topologies, although optimal, may not follow definite structures owing to the nondeterministic way they are bred. Further, they are difficult to navigate. This is not a shortcoming in applications that require broadcast from any given node to all other nodes. For example, in a supply chain it may be necessary to find the best

https://doi.org/10.25088/ComplexSystems.18.2.175

topology, which can be used to send material to all nodes from any node. Here, what is important is the efficiency with which the broadcast happens, and the existence of alternate paths, if links are unreliable. On the other hand, in applications like distributed lookups, ease of navigation is as important as efficiency and load distribution. In this section we discuss a deterministic model of building topologies that have similar properties as the bred topologies. We call this method *polygon embedding*.

5.1 Polygon Embedding

Consider Figure 6, where a polygon, a square in this case, is embedded in the outer circle. The sides of the polygon are equivalent to "chords" in the skip list topologies and provide the "long-range" connectivity.

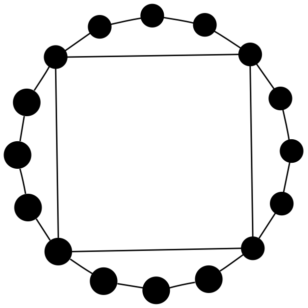


Figure 6. Embedded polygon.

In general, it can be shown that there exists a polygon of l sides, which on embedding gives a topology of diameter d such that

$$\left\lfloor \frac{n}{l} \right\rfloor + \left\lfloor \frac{l}{2} \right\rfloor - 1 \leq d \leq \left\lceil \frac{n}{l} \right\rceil + \left\lfloor \frac{l}{2} \right\rfloor - 1.$$

This expression is due to the structure of the topology, that is, a polygon embedded in a circle (Hamiltonian circuit). The polygon divides the circle into l segments, each having a length of approximately $\frac{n}{l}$: when n is perfectly divisible by l, all the segments are of length exactly $\frac{n}{l}$; otherwise, we will have $r = n \operatorname{modulo} l$ segments of length $\left\lceil \frac{n}{l} \right\rceil$ and $l - r$ segments of length $\left\lfloor \frac{n}{l} \right\rfloor$. Therefore, the maximum distance to be traversed from the source on the circle before reaching one of the polygon edges (long-range links) is $\left\lceil \frac{n}{2l} \right\rceil$. Similarly, it can be seen that the number of polygon edges to be traversed before reaching the segment

https://doi.org/10.25088/ComplexSystems.18.2.175

on the circle that has the destination node is $\left\lfloor \frac{l}{2} \right\rfloor - 1$. Again, a maximum of another $\left\lceil \frac{n}{2l} \right\rceil$ circle edges are to be traversed before reaching the destination. Thus, we get the following expressions for the diameter:

$$d = \left\lfloor \frac{n}{l} \right\rfloor + \left\lfloor \frac{l}{2} \right\rfloor - 1, \text{ when } r = 0 \text{ or } r = 1$$

$$d = \left\lceil \frac{n}{l} \right\rceil + \left\lfloor \frac{l}{2} \right\rfloor - 1, \text{ when } r > 1.$$

Upon solving these quadratic equations, we can deduce the polygon that needs to be embedded, for a given n and d

$$l = (1 + d) - \sqrt{(1 + d)^2 - 2\,n}\ .$$

From this relation, we can see that when $d < \sqrt{2\,n} - 1$, we get a complex solution. Also, we can show that d reaches a minima at $\sqrt{2\,n} - 1$. Thus, the lower bound on the diameter that can be achieved by embedding a single polygon is

$$d_{\text{lb}} = \left\lceil \sqrt{2\,n} \right\rceil - 1, \text{ when } r = 0$$

$$d_{\text{lb}} = \left\lfloor \sqrt{2\,n} \right\rfloor - 1, \text{ when } r = 1$$

$$d_{\text{lb}} = \left\lfloor \sqrt{2\,n} \right\rfloor, \text{ when } r > 1.$$

The lower bound on the diameter can be achieved by embedding a polygon of size l_{opt} given by

$$l_{\text{opt}} = \left\lceil \sqrt{2\,n} \right\rceil.$$

Further, the biggest network of diameter d_{lb} that can be constructed by embedding a polygon of l_{opt} sides is given by

$$n_{\max} = \frac{l_{\text{opt}}^2}{2}.$$

For example, given a network of 32 nodes, the optimal polygon to be embedded has eight sides. That is, embedding an octagon inside a circle of 32 nodes will reduce the diameter to 7.

https://doi.org/10.25088/ComplexSystems.18.2.175

We can also get optimal structures by embedding multiple polygons, either of the same or different sizes. We now describe two such techniques. In Figure 7, two polygons of the same size are embedded.

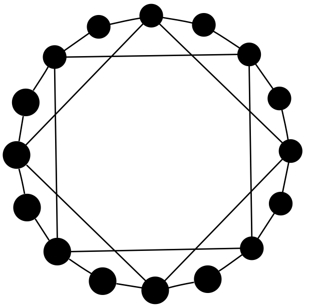


Figure 7. Two embedded polygons.

It can be deduced that the two polygons that need to be embedded, for a given n and d have l sides, where

$$l \approx d - \sqrt{d^2 - n}\ .$$

Also, the lower bound on the diameter can be achieved by embedding two equal polygons of size

$$l_{\text{opt}} = \left\lceil \sqrt{n} \right\rceil$$

and the lower bound on the diameter that can be achieved by embedding two equal polygons is

$$d_{\text{lb}} = \left\lceil \sqrt{n} \right\rceil.$$

From this relation, a network of 256 nodes can reach a diameter of 16 by embedding two hexadecagons.

Figure 8 shows an embedding in which multiple polygons are embedded by successively halving their sizes. A node is chosen as the "root" arbitrarily. This is the starting point of the embedding. In the figure, the first embedded polygon has a size of $\frac{n}{2}$. The next polygon, which has $\frac{n}{4}$ sides, is embedded starting from the root node. Successive polygons are embedded by halving until we reach the smallest polygon, a triangle or two diameters as shown in the figure.

https://doi.org/10.25088/ComplexSystems.18.2.175

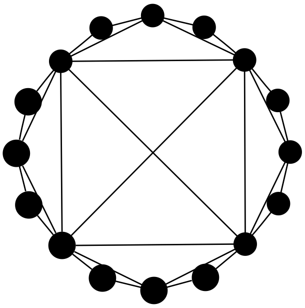


Figure 8. Polygon-halving embedding.

A polygon-halving embedding has a routing complexity that is similar to a tree rooted at the "root" node that we have selected. If the outermost embedded polygon has l sides, then the diameter of the topology can be computed from the relation

$$d = \frac{n}{l} + \log l.$$

When $l = \frac{n}{2}$, as in the previous example,

$$d = \log n + 1.$$

In the polygon embedding approach, as can be seen in Figures 7 and 8, some of the nodes have a higher degree than the others. They can be likened to "super peers". Although polygon embedding strategies are efficient, they have a low load distribution value. In environments like grids, where all nodes are not necessarily symmetric, these serve to be useful topologies. Moreover, this is not a major shortcoming as a regular graph topology can be easily achieved in a polygon embedding by replicating the structure of Figure 8 at every node. Such a topology has a diameter of $\log n$ and ρ of 1. However, the better load distribution comes at much higher infrastructure and bookkeeping costs. As a matter of fact, the topology thus obtained is equivalent to the topology of the well-known Chord [1] DHT.

5.2 Equivalent Polygon Embeddings

We consider a polygon embedding equivalent to a bred topology when at least one of its properties (namely efficiency, load distribution, or infrastructure cost) is at least as good as that of the bred topology. A bred topology can have multiple equivalent polygon embeddings with respect to one or more of these properties. Since we are interested in optimal-diameter topologies, we deal only with efficiency-equivalent topologies. Figure 9 shows examples of polygon

https://doi.org/10.25088/ComplexSystems.18.2.175

embedded topologies, which are the efficiency-equivalent topologies to some of the skip list topologies we obtained through evolution. However, they have different load distribution and cost properties.

In general, the evolved topologies exhibit better load distribution than the polygon embeddings. Further, polygon embeddings are very cost effective when the desired diameter is around the lower bounds we discussed previously. They may not be cost effective for arbitrary diameters.

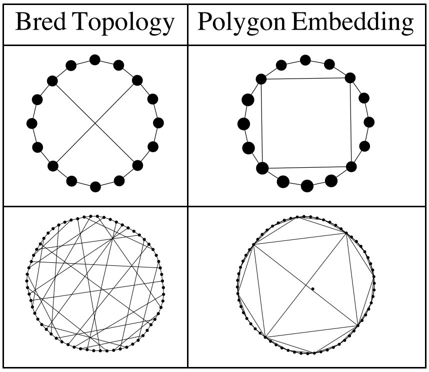


Figure 9. Efficiency-equivalent topologies.

6. Some Notes on Applications

In this section, we give a broad overview toward a few areas where our work can find application.

- *Distributed hash tables*: DHTs are the primary building blocks of data-centric overlays. Most DHTs use a distributed index to look up where a given hash bucket is located. Therefore, optimal distributed indexes are important for the success of data-centric overlays, and thereby several peer-to-peer applications.

- *Mobile collaborative applications*: We envisage that small groups of users collaborate through their mobile devices to share information that interests them. For example, users may collaborate to maintain a "global address book" of their acquaintances, or form a "social bookmarks manager" over a mobile network. Users should be able to organize their data and search efficiently in the expanded search space. Distributed indexes can be used for this purpose. In a mobile environment there are bound to be perturbations. Knowledge of different optimal topologies helps here. For example, if some nodes or edges are lost, the rest of the nodes can "snap" to the nearest optimal topology.

- *Information architecture*: Ease of navigation is an important design requirement for information systems. Users should be able to reach the information they need in a simple and efficient manner. In other words, information should be arranged in a manner that facilitates simple and

https://doi.org/10.25088/ComplexSystems.18.2.175

efficient navigation. Again, an optimal distributed index topology, such as an optimal hyperlink graph topology, is useful here.

7. Conclusions and Future Work

We propose to develop general principles for designing distributed indexes. The proposed approach for optimal topology design provides a generic perspective to a class of problems involving topology design under different conditions. While some of the results in the current work (e.g., the formation of trees, stars, and circles) appear obvious in retrospect, we have not come across any theories that predict these topologies given initial conditions. Also, another important insight that we found is the appearance of at least one Hamiltonian circuit in topologies whose connectivity crosses the bare minimum threshold of $n-1$ edges. Although it is intuitively apparent, we have not found a proof for the requirement of a Hamiltonian cycle. Developing a proof for this is an important part of the future work.

Knowledge of optimal distributed index topologies in a generic sense helps us in classifying them and mapping them onto classes of real-world applications that require optimal distributed indexes. It also helps in designing snapping algorithms that snap back to the nearest optimal topology in the face of perturbations. This can be crucial to the design of high-performance mobile and ad hoc data-centric networks.

Presently we are working with efficiency, load distribution, and cost as the constraints. Future work includes identifying more constraints such as robustness and mobility, and incorporating them into our model. Future work also includes developing algorithms. By using such algorithms, a topology can snap on to the nearest known optimal topology, in a distributed manner.

Acknowledgments

We thank Infosys Technologies Limited for granting a Ph.D. scholarship to the first author to conduct this work. P. C. P. Bhatt's insightful and timely inputs are much appreciated. Thanks are also due to Reshma Ratnani and Roshini T. Raj, who developed a topology visualization and analysis tool (TopAZ).

https://doi.org/10.25088/ComplexSystems.18.2.175

References

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ATAPCC'01)*, San Diego, New York: ACM, 2001 pp. 149-160.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01)*, Heidelberg (R. Guerraoui, ed.), *Lecture Notes in Computer Science*, New York: Springer-Verlag, 2001 pp. 329-351.

[3] M. F. Kaashoek and D. R. Karger, "Koorde: A Simple Degree Optimal Distributed Hash Table," in *Proceedings of the Second International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Berkeley (F. Kaashoek and I. Stoica, eds.), *Peer-to-Peer Systems II, Lecture Notes in Computer Science*, **2735/2003**, New York: Springer-Verlag, 2003 pp. 98-107.

[4] P. Fraigniaud and P. Gauron, "An Overview of the Content-Addressable Network D2B," in *Proceedings of the Twenty-Second Annual ACM Symposium on Principles of Distributed Computing (PODC'03)*, Boston, New York: ACM, 2003 p. 151.
doi.acm.org/10.1145/872035.872056.

[5] D. Loguinov, J. Casas, and X. Wang, "Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience," *IEEE/ACM Transactions on Networking*, **13**(5), 2005 pp. 1107-1120.

[6] S. Ratnasamy, S. Shenker, and I. Stoica, "Routing Algorithms for DHTs: Some Open Questions," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA (P. Druschel, F. Kaashoek, and A. Rowstrom, eds.), *Peer to Peer Systems*, *Lecture Notes in Computer Science*, **2429/2002,** Heidelberg: Springer, 2002 pp. 45-52.

[7] J. Xu, A. Kumar, and X. Yu, "On the Fundamental Tradeoffs between Routing Table Size and Network Diameter in Peer-to-Peer Networks," *IEEE Journal on Selected Areas in Communication (J-SAC): Recent Advances in Service Overlay Networks*, **22**(1), 2004 pp. 151-163.

[8] V. Venkatasubramanian, S. R. Katare, P. R. Patkar, and F. Mu, "Spontaneous Emergence of Complex Optimal Networks through Evolutionary Computation," *Computers and Chemical Engineering*, **28**(9), 2004 pp. 1789-1798. arXiv:nlin/0402046v1 [nlin.AO].

[9] W. G. Bridges and S. Toueg, "On the Impossibility of Directed Moore Graphs," *Journal of Combinatorial Theory*, Series B, **29**(3), 1980 pp. 339-341. doi:10.1016/0095-8956(80)90091-X.

[10] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "HyperCuP—Hypercubes, Ontologies, and Efficient Search on P2P Networks," in *Proceedings of the First Workshop on Agents and Peer-to-Peer Computing (AP2PC'02)*, Bologna, Italy (G. Moro and M. Koubaraks, eds.), *Agents and Peer-to-Peer Computing*, *Lecture Notes in Computer Science*, **2530/2008**, New York: Springer, 2008 pp. 133-134.

https://doi.org/10.25088/ComplexSystems.18.2.175

[11] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A Scalable and Dynamic Emulation of the Butterfly," in *Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing (PODC'02)*, Monterey, CA, New York: ACM, 2002 pp. 183-192. doi.acm.org/10.1145/571825.571857.

[12] A. Kumar, S. Merugu, J. Xu, and X. Yu, "Ulysses: A Robust, Low-Diameter, Low-Latency Peer-to-Peer Network," in *Proceedings of the Eleventh IEEE International Conference on Network Protocols (ICNP'03)*, Atlanta, New York: IEEE Press, 2003 pp. 258-267. doi.10.1109/ICNP.2003.1249776. *European Transactions on Telecommunications*, **15**(6), 2003 pp. 571-587 (revised version).

[13] M. J. Freedman and R. Vingralek, "Efficient Peer-to-Peer Lookup Based on a Distributed Trie," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA (P. Druschel, F. Kaashoek, and A. Rowstrom, eds.), *Peer to Peer Systems*, *Lecture Notes in Computer Science*, **2429/2002,** Heidelberg: Springer, 2002 pp. 66-75.

[14] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing Nearby Copies of Replicated Objects in a Distributed Environment," in *Proceedings of the Ninth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'97)*, Newport, RI, New York: ACM, 1997 pp. 311-320. doi.acm.org/10.1145/258492.258523.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of the Conferenece on Applications, Technologies, Architectures, and Protocols for Computer Communications (ATAPCC'01)*, San Diego, New York: ACM, 2001 pp. 161-172. doi.acm.org/10.1145/383059.383072.

[16] G. Pandurangan, P. Raghavan, and E. Upfal, "Building Low-Diameter P2P Networks," in *Proceedings of the Forty-Second IEEE Symposium on Foundations of Computer Science (FOCS'01)*, Las Vegas, Washington, DC: IEEE Computer Society, 2001 pp. 492-499.

[17] C. Law and K.-Y. Siu, "Distributed Construction of Random Expander Graphs," in *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, San Francisco, New York: IEEE Press, 2003.

[18] J. Kleinberg, "The Small World Phenomenon: An Algorithmic Perspective," in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC'00)*, Portland, OR, New York: ACM, 2000 pp. 163-170. doi.acm.org/10.1145/335305.335325.

[19] G. S. Manku, M. Bawa, and P. Raghavan, "Symphony: Distributed Hashing in a Small World," in *Proceedings of the Fourth USENIX Symposium on Internet Technologies and Systems (USITS'03)*, Seattle, USENIX Association, Berkeley, CA, 2003 pp. 127-140.

[20] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The Impact of DHT Routing Geometry on Resilience and Proximity," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM'03)*, Karlsruhe, Germany, New York: ACM, 2003 pp. 381-394. dx.doi.org/10.1145/863955.863998.

[21] V. Venkatasubramanian, "A Theory of Design of Complex Teleological Systems: Unifying the Darwinian and Boltzmannian Perspectives: Essays and Commentaries," *Complexity*, **12**(3), 2007 pp. 14-21.
doi .10.1002/cplx.v12:3.

https://doi.org/10.25088/ComplexSystems.18.2.175