Topology as a Factor in Overlay Networks Designed to Support Dynamic Systems Modeling

Abbe Mowshowitz[1], Akira Kawaguchi[1], and Masato Tsuru[2]
[1] Department of Computer Science, The City College of New York, New York, U.S.A.
amowshowitz@ccny.cuny.edu, akawaguchi@ccny.cuny.edu
[2] Kyushu Institute of Technology, Iizuka, Japan
tsuru@csn.kyutech.ac.jp

Abstract. Overlay networks are logical networks embedded in physical substrate networks. They are useful for supporting specialized applications involving computation and information exchange among subsets of users. This paper examines the characteristics of overlays that make them suitable for different applications in dynamic, distributed database environments. One such characteristic is the ease with which distances between nodes can be calculated. Examples of overlay graph structures that exhibit certain desirable characteristics include the hypercube, toroidal grid graph and the Kautz graph. Applications involving mobile elements are examined, and a system designed to support comparative analysis of the performance of different overlay structures is discussed.

1 Introduction

Characteristics of network topology can have significant effects on network performance. This is especially true for a highly distributed data network such as a 5G platform. Distance between nodes, for example, can affect latency, and may also play a role in traffic congestion if large amounts of data must be moved in response to queries. Previous research proved in theory that traffic congestion in overlay networks can be reduced by engineering the network to conform to a hypercube structure [1,6,10,11,16]. In a current project we are implementing a hypercube overlay by means of software defined networking. The aim of this experimental research is to provide a practical demonstration of increased querying efficiency in distributed database systems, achieved by taking account of distance between nodes [14].

In this paper we generalize the previous work through a systematic examination of the advantages and disadvantages of different overlay network topologies for different applications. The principal contribution of this work lies in demonstrating the utility of particular layover topologies designed to support dynamic systems modeling.

2 Overlay Network

An *overlay network* is a virtual or logical network embedded in a physical substrate such as a WAN, MAN or the Internet [15]. To illustrate the relationship between overlay and substrate, suppose nodes *a* and *b* are connected by a path *P* of length *k* in the overlay. If a message is sent from *a* to *b* in the overlay, it could follow an entirely different path *Q* in the substrate whose length could be less than or greater than *k*. If an overlay is well aligned with the substrate, the difference in path lengths will be minimal. Overlay networks are typically used to establish communication and information exchange among a subset of substrate users for specialized activities.

3 Network Topologies and Graph Models of Networks

Network structure can be characterized abstractly by means of graphs. Our concern here is with dynamically changing networks, whose structure varies over time due to the entrance of new nodes and disappearance of existing ones [1]. If there are no constraints on the evolution of such a network, the structure is in effect a random graph. However, evolving networks can be structured to conform to predetermined graph types. The rationale for structuring growth is that the resulting overlay network will resemble a particular type of graph with known properties. For example, if distance between nodes is important, a graph structure can be chosen that has a small diameter, and low average distance between nodes [8]. Moreover, with some graphs (e.g., hypercube), the node labels allow for easy determination of distance and computation of paths between pairs of nodes [16]. This property provides a systematic way of optimizing distributed data exchange so as to minimize overall data transfer on the network. Broadcasting offers another example of the potential utility of a structured overlay. If the graph associated with an overlay has a Hamiltonian circuit, that structure could be exploited to establish an efficient algorithm for broadcasting.

One structure long used in computational modeling is the *mesh* or *grid graph*, also known as a *lattice graph* [18]. The simplest type of this structure is the *square grid graph* which is defined as the Cartesian product (P_m X P_n) of two paths P_m and P_n , having m and n vertices (m-1 and n-1 edges), respectively. In a notable application of this graph, the vertices correspond to processors and edges represent direct communication links between processors. Path lengths in P_m X P_n can be as high as (m-1) + (n-1), the diameter of the graph. If a large amount of data has to be shared between processors separated by considerable distance, network performance can be compromised. By contrast, the hypercube H_k with 2^k vertices has diameter k and offers an average path length smaller than the square grid graph. For example, if m and n are 32 and k is 10, both graphs have 1024 vertices, but the diameters are far apart: the hypercube has diameter 10, while the square grid graph has diameter 62.

One additional property of the hypercube is of special interest, namely the labels associated with its vertices. In fact, the labels can be used to define the hypercube: H_k = (V,E), where V is a set of 2^k vertices, and E is a set of edges. Each vertex has a unique binary k-sequence label, and e = uv ∈ E if the labels of u and v differ in exactly one position. The distance between two vertices is the Hamming distance between their respective labels. Thus, structuring a layover network as a hypercube makes distance computation almost immediate, whereas in the random graph case it is necessary to exchange messages between nodes to determine the network distance between them.

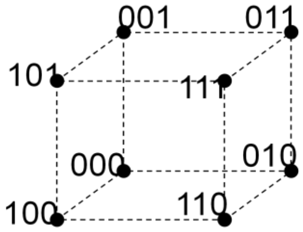
4 What Constitutes a Good Overlay Network Topology?

Some network properties such as fault tolerance or resistance to attack, as measured, for example, by the difficulty of breaking a network into isolated components, are critical for all network applications. However, a 'good topology' is generally one that answers to the needs of a particular type of application. We have already called attention to diameter and average path length as important properties for dynamic, distributed database operations, since a small diameter and low average path length can help to reduce message traffic and thus improve latency and throughput. Unlike its unstructured cousin, a structured overlay must be maintained which requires effort and entails cost. This consideration leads us to examine the tradeoffs between desirable overlay topologies and the cost of maintaining such topologies.

To illustrate the trade off issues we will discuss three different overlay topologies that can be used to support dynamic, distributed database operations. In addition we will examine an application involving mobile devices. Distributed database operations call for layover networks in which distance between nodes can be determined easily. Other applications may require close alignment of real world features, such as physical proximity, with network properties; and maintenance costs of a given overlay topology may be a critical concern.

4.1 Hypercube

As noted earlier, the hypercube (see Fig. 1) is well suited to support distributed database operations inasmuch as a simple bit vector operation on pairs of labels can b



Labels 000 to 111, nodes differing in one bit are adjacent

Fig. 1. Hypercube

be used to compute distance. Moreover paths can be constructed easily using the node labels. The cost of maintaining the hypercube overlay is more than offset by the savings in bandwidth utilization if the overlay network is relatively stable under conditions of intensive querying. If on the other hand, the rate of change in the network caused by nodes entering and leaving is high, maintenance costs can become excessive. The cost of maintaining a hypercube overlay structure stems in part from the relatively high edge density of this type of graph. The hypercube H_k has k/2 edges per node. Connections between nodes in an overlay network must be checked to be sure they are intact, so a high ratio of edges to nodes incurs considerable maintenance cost. This high ratio, making the hypercube robust and resistant to attack, is thus a negative for maintenance.

4.2 Toroidal Grid Graph

A topology that also has the virtue of natural labels that afford simple inter-node distance calculation is the *toroidal grid graph*, so called because it can be embedded in a torus. This graph (see Fig. 2) is the Cartesian product C_m X C_n of cycles C_m, C_n of lengths m and n, respectively; it has mn nodes, 2mn edges, constant edge density 2, and is regular of degree 4. Compared with the hypercube, the low edge density offers a reduction in the cost of edge connection maintenance if it is used as an overlay network. The Cartesian product offers a natural edge labeling scheme, since nodes have the form (i,j), and the distance d[(i,j),(p,q)] between nodes (i,j) and (p,q) is given by the formula $d[(i,j),(p,q)] = d_{Cm}(i,p)$, if j=q; or $d_{Cn}(j,q)$, if i=p; or $d_{Cm}(i,p) + d_{Cn}(j,q)$, otherwise, where $d_{Ck}(x,y) = |x-y|$ if $|x-y| \leq \lfloor n/2 \rfloor$, or else $d(x,y) = k - |i-j|$. So, the toroidal grid graph can be used to support distributed database operations, at a lower cost than the hypercube. The downside of the relatively low edge density is greater vulnerability to attack, since fewer elements than in the hypercube are needed to break the network into disjoint pieces that cannot communicate with each other.

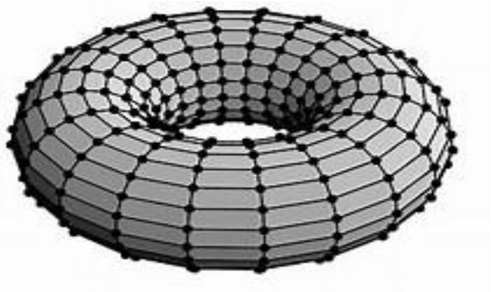


Fig. 2. Toroidal Grid Graph [8]

Note that every Cartesian product affords a natural labeling scheme that can be used to compute distance easily, assuming the constituent graphs in the product also have an easy to compute distance metric. The different Cartesian product graphs constitute an infinite class of potential layover networks that could be used in applications involving distributed database operations. Interestingly, the hypercube too can be defined recursively using the Cartesian product. Other graph product operations (e.g., lexicographic product) could also serve as potential layover topologies.

4.3 Kautz Graph

This graph (see Fig. 3), like the hypercube and the toroidal grid graph, has been used in the design and analysis of interconnection networks, linking processors or switching elements [17]. The undirected Kautz graph is typically defined in two steps, first as a directed graph, and then with a straightforward modification, it is turned into an undirected graph. Node labels play a critical role, just as in the definition of the hypercube. A vertex of the directed Kautz graph K(d,n) has as label a length n sequence of the form $x_1,x_2,\ldots,x_n$ where the x_i take values in the set $S = \{0,1,\ldots,d\}$ for some integer $d>0$, subject to the condition that $x_{i+1} \neq x_i$. A vertex u is adjacent to vertex v if the label of v is a one position right shift of u's label with a new value from the set S tacked on to the end. K(d,n) is d-regular (i.e., indegree = outdegree = d), with $(d+1)d^{n-1}$ vertices and $(d+1)d^{n}$ edges. Like the toroidal grid graph K(b,n) has a fixed number d of paths between vertices. The undirected Kautz graph UK(d,n) is obtained from K(d,n) by deleting the orientation of all directed edges and keeping one edge of each pair of multiple (bidirectional) edges. Like the toroidal grid graph C_m X C_n, UK(d,k) graph has a constant edge density approximately equal to d. UK(d,k) is fault tolerant, but not as resistant to attack as the hypercube H_k. With a relatively small diameter (n), and relatively high connectivity (2d- 1), UK(d,n) is desirable as a network topology [2,3]. However, inter-node distance computation based on the labels is more complex than in the hypercube or the toroidal grid graph, which makes this graph

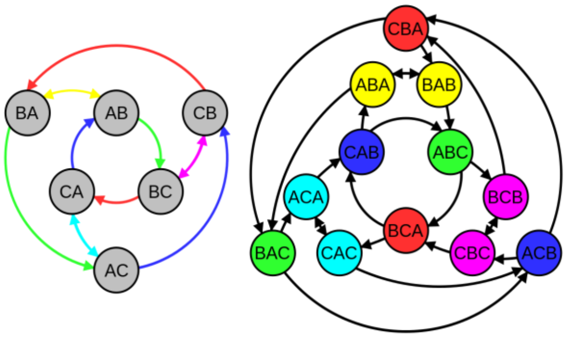


Fig. 3. Directed Kautz graphs with |S| = 3 and string length 2 and 3 [19]

topology less attractive for applications in which distance between nodes plays a critical role.

The three graphs discussed here do not exhaust the possibilities. Graph operations such as tensor product and composition could also be used to design overlay topologies. The graphs examined here are offered as examples for the purpose of demonstrating the issues of concern in evaluating topologies for overlay networks designed to support particular applications.

5 Example of an Overlay Topology to Support a Mobile Application

Thus far we have emphasized the need for labeling schemes that permit simple computation of inter-node distances in overlay networks. Other properties may be equally important in some applications. Another important consideration has to do with maintenance of the overlay topology. Connections between nodes must be checked and sometimes re-established, so the edge density of an overlay network may be an important parameter. Consider a smart environment application in which cars circulating in a city need to share information and to determine which other cars are nearby. For simplicity we speak of cars without mentioning drivers explicitly, although the latter are the agents utilizing the information being exchanged. The scheme proposed here as an illustration is different from a Mobile Ad Hoc Network (MANET). We assume the cars can connect to a certain edge-based network (or possibly a WAN or the Internet) and make use of an overlay network embedded in that substrate to exchange messages with other cars (see Fig. 4). To facilitate communication and message passing between neighbors, the messaging system should take account of both network distance between nodes representing cars in the overlay and distance between cars in real physical space. Ideally, the assignment of nodes to cars would group neighboring cars close together in the overlay network.

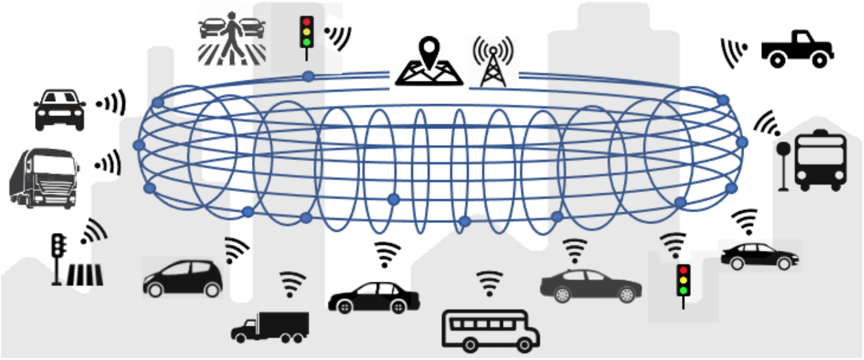


Fig. 4. A Mobile Application Utilizing a Toroidal Grid Graph Overlay

An important requirement in this case is a close alignment between network distance and physical distance. The proposed scheme uses the toroidal grid graph and works as follows. A car announces its intention to join the overlay network by broadcasting a message which includes its GPS location. Nodes currently in the network receiving the message will respond by providing their GPS location, their node label, and a free label in their immediate neighborhood. The new entrant will compare its GPS location with those of the first, say three, responding nodes and accept the label sent by the node closest to itself in both real space and the overlay network based on a fitness function of both parameters. Taking account of the movement of cars in the city, and to assure a close alignment of network and physical distance, a car would remain connected to the overlay network for only a specified period of time. This means that at the end of the time interval the car would disconnect and repeat the joining operation again, presumably from a new physical location. To determine all the cars within a given radius of its current location, a car would broadcast a request giving its GPS location, overlay network label, and specification of the target distance from its current location. A receiving car would compare its current location with that of the calling car and respond appropriately. The toroidal grid graph is a suitable topology for this application because its vertex labels permit easy determination of network distance between nodes, and its edge density is relatively low, thus keeping the maintenance cost relatively low. Hypercube and Kautz graphs could also be used, which would allow for comparative analysis of the advantages and disadvantages of different topologies.

This example of a possible application is meant to illustrate the importance of topology as a factor in overlay network design. The element of mobility draws attention of the relationship between physical and network distance. In practice, a number of issues would have to be addressed to make the application work. In particular, the substrate network would have to be specified as well as the precise method of accessing the substrate.

The conceptual design described above could in principle be adapted to any setting in which mobile devices connected to some substrate need to share information. Air traffic control, for example, could be managed in this way, if aircraft were supplied with the requisite instruments needed to connect to the substrate. This approach could offer an effective and cheaper way of managing control than is currently available. Another interesting area involves the spread of contagious diseases. Treating contagion as a kind of message passing among mobile individuals is an innovative way of modeling epidemics that could offer analytic tools for designing effective interventions [4].

6 Work in Progress

We are implementing a comprehensive simulation system in support of the comparative investigation of overlay network topologies. The design is based on multiple open-source, heterogeneous, relational database systems such as MariaDB, PostgreSQL, Firebird, SQLite3. These systems are built on top of OpenFlow environment embedded with the software-oriented Address Resolution Protocol. To insure portability of the simulated environment, the experimental system makes use of the Ubuntu 18.04 operating system built with a Mininet virtual net enhancement (see Fig. 5). Initially, the system will create a static hypercube to accommodate a predetermined number of nodes, and will activate an overlay structure by accessing a designated IP address that hosts a specific database system. Our aim is to build a working and practical demonstration of the advantages of overlay networks.

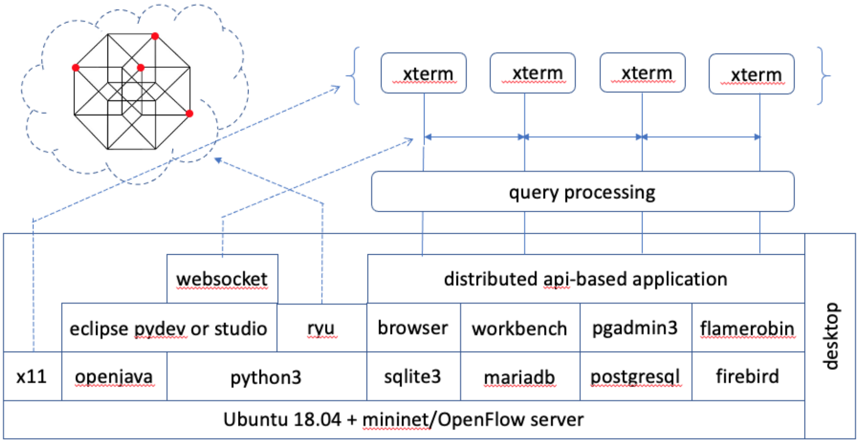


Fig.5. Simulated Software Environment

The experiments will pay special attention to the data requirements of users. To this end, we are building several distributed query processing applications designed to gather data from a publicly available database system, namely New York City OpenData. Collections of interest in this system include records of parking violations, housing litigation, arrests, complaints, etc. Currently, the data gathered from New York City OpenData, ranging in size from a hundred MB to one GB) are placed in a centralized repository. For this application the data sets are divided it into several relational databases corresponding to the grouping of district IDs, precincts, building IDs, etc. These databases are connected through WebSockets, and therefore they can process bidirectional data transfers. Critical parameters of query performance are response time and the total amount of data transmitted on the network. Comparison of performance between a hypercube overlay network and a set of randomly built networks is underway.

7 Conclusion

Overlays are useful tools for connecting subgroups of network users for applications involving dynamic systems, and the topology of an overlay is a critical success factor in such applications. Further research is needed to categorize overlay network topologies based on features needed to support particular applications. Such a categorization of topologies, based on selected features of interest, investigated both analytically and experimentally, would allow for selecting the best topology for any given dynamic modeling application.

References

[1] Bent, G., Dantressangle, P., Vyvyan, D., Mowshowitz, A. and Mitsou, V. A dynamic distributed federated database. *Annual Conference of the International Technology Alliance*, Imperial College-London, 2008.

[2] Bermond, J.-C., Homobono, N., and Peyrat, C. Connectivity of Kautz networks. *Discrete Mathematics* **114**, 1993, pp. 51-62.

[3] Du, D.Z., Hsu, D.F., and Lyuu, Y.D. On the diameter vulnerability of Kautz digraphs. *Discrete Mathematics* **151**, 1996, pp. 81-85

[4] Enright, J. and Kao, R.R. Epidemics on dynamic networks. *Epidemics* **24**, 2018, pp. 88–97.

[5] Imrich, W. and Klavzar, S. *Product Graphs: Structure and Recognition*. New York: Wiley, 2000.

[6] Kawaguchi, A., Mowshowitz, A., Nagel, A., Toce, A., Bent, G., Stone, P., and Dantressangle, P. A model of query performance in Dynamic Distributed Federated Databases taking account of network topology. *Annual Conference of the International Technology Alliance*, Southampton, UK, September 2012.

[7] Marquez, A., de Mier, A., Noy, M. and Revuelta, M.P. Locally grid graphs: classification and Tutte Uniqueness. *Discrete Mathematics* **266**, 2003, pp. 327 – 352.

[8] MathOverflow. Torus graph dynamics, 2021. https://mathoverflow.net/questions/270761/torus-graph-dynamics

[9] Mowshowitz, A., Mitsou, V., Bent, G. Evolving networks and their vulnerabilities. In: Dehmer, M., *et al*. editors, *Modern and Interdisciplinary Problems in Network Science: A Translational Research Perspective*, CRC Press, 2018, pp. 173-192.

[10]Mowshowitz, A., Kawaguchi, A., Toce, A., Nagel, A., Stone, P., Dantressangle, P., and Bent, G. Network topology as a cost factor in query optimization . *Annual Conference of the International Technology Alliance*, Adelphi, MD, 2011.

[11]Mowshowitz, A., Kawaguchi, A., Toce, A., Nagel, A., Bent, G., Stone, P., and Dantressangle, P. Query optimization in a distributed hypercube database. *Annual Conference of the International Technology Alliance*, Imperial College-London, September 2010.

[12]Mowshowitz, A., Bent, G., and Dantressangle, P. Aligning network structures: embedding a logical dynamic distributed database in a MANET. *Annual Conference of the International Technology Alliance*, University of Maryland, 2009.

[13]Mowshowitz, A. and Bent, G. Formal properties of distributed database networks. *Annual Conference of the International Technology Alliance*, University of Maryland, 2007.

[14]Sadaawi, T., Kawaguchu, A., Lee, M., and Mowshowitz, A. Secure resilient edge cloud designed network. *IEICE Transactions on Communications*, in press.

[15]*Tarkoma, S. Overlay Networks: Toward Information Networking. CRS Press, 2010.*

[16]Toce, A., Mowshowitz, A., Kawaguchi, A., Stone, P., Dantressangle, P., and Bent, G. An efficient hypercube labeling scheme for dynamic Peer-to-Peer networks. *J. Parallel Distrib. Comput.* **102**, 2017, pp. 186-198.

[17]Wang, S. and Lin, S. The k-restricted edge connectivity of undirected Kautz graphs. Discrete Mathematics **309**, 2009, pp. 4649-4652.

[18]Wikipedia. Lattice graph, 2020. https://en.wikipedia.org/wiki/Lattice_graph

[19]Wikipedia. Kautz graph, 2021. https://en.wikipedia.org/wiki/Kautz_graph