ANALYZING THE IMPACT OF LOCAL PERTURBATIONS OF NETWORK TOPOLOGIES AT THE APPLICATION-LEVEL

BY VINCENT J. MATOSSIAN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Manish Parashar

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE DISSERTATION

Analyzing the Impact of Local Perturbations of Network Topologies at the Application-level

by Vincent J. Matossian

Dissertation Director: Professor Manish Parashar

Networked systems are continuously growing in scale and complexity. The technical and policy engineering challenges introduced by such a fast growth are currently addressed locally, with limited understanding of their impact on the whole. Such approaches are becoming impractical and insufficient. Next-generation networks need to address these issues by deploying adaptive and self-managing protocols and mechanisms to relax the persistent need for human-driven management. However, achieving these objectives requires conceptual, physical, and logistical modifications to existing systems and protocols. To this end, the traditional top-down approach to network and application design needs to be supplemented by understanding the bottom-up nature of evolving real-world networks.

A critical issue that is significantly impacting computer networks and applications is the absence of an in-depth understanding and lack of control over the structural properties, i.e., topology, of large networks. Network topologies define the link relationships

between the nodes in the network, and have a direct impact on the performance, resilience, and security of distributed applications. Large scale networks such as the Internet are the result of a time evolving process in which nodes and links between nodes are added, removed, and reconfigured dynamically. This dynamic process takes place in a decentralized manner during which nodes make local adaptations and reconfiguration decisions that optimize local properties. As a result, these local perturbations yield an emergent network that is often unstructured and complex, and have implications at the application-level, particularly impacting routing, search, robustness, and clustering. Understanding the structures emerging out of these adaptations is a complex problem part of the science and study of complexity theory and complex adaptive systems. Tackling this complex problem requires first, identifying canonical metrics to quantify the network topology and second, analyzing the impact of local perturbations of these metrics on the resulting network topology.

This thesis identifies three local metrics, transitivity, assortativity, and entropy, and analyzes the impact of their perturbation on the applications of routing, search, robustness, and clustering. The local metric of network entropy is identified as a useful information theoretic measure of homogeneity of a network neighborhood degree. The metric is further used to derive a novel mechanism of clustering detection of the network topology. The overall objective of this thesis is to investigate metrics and mechanisms to better understand the evolution of the network topology and its impact on application-level functionality. The approach is based on concepts of emergence, self-organization and graph theory, and has three key aspects: (1) the identification of canonical local and global graph metrics; (2) the quantitative analysis of the impact of local perturbations on global properties; and (3) the application of the local to global mapping on the problems of routing, search, robustness, and clustering. Adaptations are performed in a decentralized manner in which local nodes use local information to add, remove, or rewire an edge to evolve the topology.

Simulations based on annealing optimization are conducted to empirically determine the optimal bounds of the network structures for the selected metrics on selected networks. Further experiments on two modeled networks, random and power-law degree distributed, and two real-world networks, the Gnutella and Canadian Autonomous System networks, show that the impact of optimizing networks with fixed degree distribution on local metrics yield networks with routing, search, robustness, and clustering that are tightly dependent on the network's degree distribution. A key outcome of this thesis is the identification of network entropy minimization as a useful local rewiring strategy to decrease average path length and search cost, while homogenizing the size of network clusters and having a low impact on robustness when applied to power-law degree distributed networks that prevail in real-world networks.

Preface

Paradoxically, in most publications conclusions are written first and the preface last. This gives a unique opportunity for the author to express, in the very first pages, the lessons learned in the process. I take this opportunity to share how this work came to be and what sustained me throughout.

Upon first joining Professor Parashar's Applied Software Systems lab, I developed three-tiered architectures to provide physicists with online tools to interact with high-performance scientific applications. While the three-tiered architectures were already quite challenging, they extended into *n-tiered* ($n >> 3$) Peer-to-Peer network architectures. These systems considered a large number of interacting compute nodes and faced many fundamental problems in distributed systems. I devoted my Masters thesis to the development of a peer-to-peer messaging library to decouple monolithic legacy scientific applications into autonomically interacting peer services. My fascination for networks was strengthened by the conceptual relationships between P2P networks and other large-scale networks, in particular social networks. I participated in the development of the overlay network associative messaging for data mining that lead me to question, *how do network topologies (self-) organize?*

What supported me throughout this work, besides Professor Parashar's unconditional

support, are two ideas:

- The first is the famous motto *Think Globally and Act locally*, which is an inspiring call aimed at raising our awareness for the environment and the world by adding accounting into the individuals in a population.

- The second is the *Pareto principle* or *80-20 rule* that in essence says that a distribution is skewed such that 80% of it is represented by only 20% of a population. Or for that matter, that only 20% of the people may really be thinking globally.

These contrasting statements make me wonder, how would the world be if everyone disproved the Pareto principle and *did think globally*? The answer, I think, is yet to be determined...

Acknowledgements

It is difficult to find balance in thankfulness. It is a religion. One either surrenders to the vulnerability of life's essence, or feels confident that all that is is the fruit of their labor alone. To thank others is to recognize their contribution and help in a time of development of the self. Even though I do not know whether my achievement will take me towards good or bad outcomes, I have been supported on this exciting and stimulating adventure in many more ways than I can express.

I thank my adviser, Professor Manish Parashar, for his continuous support throughout the long years that this work has taken to reach maturity. his excitement for new ideas and technologies made our discussions take unforeseen directions that often lead to abstract and fundamental problems, the essence of research. Special thanks to Professor Ivan Marsic for stimulating discussions that made me realize how much research lies on a fine line saddled between strength and vulnerability. Many thanks to members of my committee, Professor Yanyong Zhang, Professor Marco Gruteser, and Professor Scott Klasky for taking time in their busy schedule to contribute to my thesis. Thanks to all the resourceful staff at CAIP, Bill Kish, James Chun, Stephen Carter, Judy Pellicane, Carmen Elsabee, to name a few, for making all administrative and technical issues non-issues... I do recognize how much frustration you all have spared me. Special thanks to Bill Kish for his love of music that have many times revived my own passion for it. I wish to thank Gábor Csárdi for developing the igraph library, making it available for free, and for his invaluable and prompt help and support.

I thank Scott Page, John Miller, and all the participants in the 2007 Computational

Social Science Workshop at the Santa Fe Institute for their engaging research interests and warm personalities. It was an invaluable experience that gave me a context and frame of work that supported me in more ways that I can express.

I warmly thank my mother Olga and brother Léonardo, for encouraging me to travel far away from them in search of myself, and for always supporting my views and interests. Thanks to my friend Phillip Stanley-Marbell for the irreplaceable discussions on everything from cooking to compiling that were always sustained with the feeling that the limits of curiosity grow exponentially as one develops interests.

Finally, and most importantly, I want to thank my dear and beautiful Kym for giving me the most precious gift, that of learning to put life in a new perspective every day.

Dedication

To Léonardo, to my mother Olga, and to the memory of my father.

Table of Contents

List of Tables

List of Figures

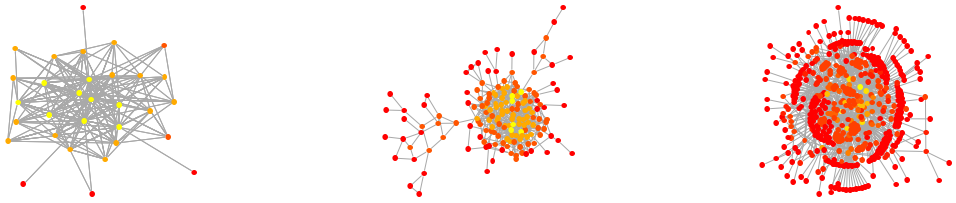
Chapter 1
Introduction

1.1 Motivation

In the past decade, the study of the dynamics of large-scale computer networks have been added to a list of open problems related to the study of complex systems. Problems in this category include, for example, protein interaction maps in Biology, citation networks in Social Science, the evolution of Autonomous Systems interconnects in Computer Networks. Figures 1.1(a) 1.1(b) 1.1(c) show force-based graph layout [2] representations of such networks. While these figures do not provide any scientific evidence as such, one can visually note the absence of any apparent pattern or structure between the elements of the graph. The common thread across all these problems is the underlying complex web of links that tie the elements together, forming a whole that is greater than the sum of its parts. The dynamic behavior is *complex* rather than *complicated* due to the interdependencies between elements that when perturbed, even so slightly, can render the system inoperable [3].

Such complexity is becoming more apparent on large-scale computer networks such as the Internet, possibly due to the intricate multi-layered structure of protocols, and hardware heterogeneity, that form a complex chain of dependencies, in which a change in one part can cause large and unexpected deviations in another. Attempts at understanding the structure of the Internet are only recent. However, the necessity to reach this understanding is ever greater, as problems across a multitude of disciplines, from

science and humanities to business are increasingly depending on large computing platforms, which are expected to support reliable and secure heterogeneous applications. For example, as of September 2007 [1], the distributed computing framework *Berkeley Open Infrastructure for Network Computing* (BOINC) [4] performs an average of 573 TeraFlop/s. Which is currently more computing than the most powerful existing supercomputer, the IBM BlueGene/P with a theoretical peak performance of 360 TeraFlop/s. As another example, in the first quarter of 2007 retail sales in electronic commerce conducted on the Internet represented 3.3% of the total retails sales in the USA[2], a value that has been growing at a constant rate since the adoption of the Internet as a business platform in the late 1990's. As the demand on computing power increases and the expectations on interoperability and fault-tolerance grow, the need to understand and manage the networks become increasingly critical to the future of such applications.



(a) (b) (c)

Figure 1.1: (a) Protein Interaction Maps. 31 Nodes using compressed view from a 1458 node network [4]. (b) Citeseer coauthor network. Top 200 authors. Data compiled by the author. (c) Canadian Internet's Autonomous Systems. 496 Nodes. Data compiled by the author.

The motivation underlying the research presented in this thesis draws from the following observations:

The network topology defines the link relationships between nodes that represent network elements, and links that describe the *"who knows who"* relation between

[1] see http://boincstats.com/stats/project_graph.php

[2] see http://www.census.gov/mrts/www/data/html/07Q1.html

nodes.

The representation of networks as graphs of nodes and links enables the application of the mathematical tool of graph theory to the analysis of networks.

Using graph theory, the structural properties of the network topology can be characterized using metrics that quantify the topology and qualitatively correlate the topology to a class of applications.

Large-scale networks such as the Internet are the result of a time evolving process in which nodes and links between nodes are added, removed, and reconfigured dynamically. This dynamic process takes place in a decentralized manner during which nodes make local adaptations and reconfiguration decisions that optimize local properties.

As a result, these local perturbations yield an emergent network that is often unstructured and complex, and have implications at the application-level.

Understanding the impact of local adaptations on global structures is a complex problem, part of the science and study of complexity theory and complex adaptive systems.

However, with networks such as the Internet growing in scale, demand, and expectations, and with the highly dynamic and quasi-instantaneous software-level rewirings offered by overlay networks, it is becoming increasingly important to address this problem so that next-generation networks can be better managed and understood.

To tackle this complex problem that has limited theoretical results [5] for real-world networks, one has to take inductive, empirical, and exploratory steps, first by identifying metrics to quantify the topology, and second by analyzing the impact of local perturbations of these metrics on the resulting networks.

This thesis is the result of one such exploration into the identification of local metrics, namely transitivity, assortativity, and entropy, and the analysis of the impact of their perturbation at the application-level for network topologies with given degree distributions. The application-level properties considered are fundamental building blocks of any distributed application and consist of:

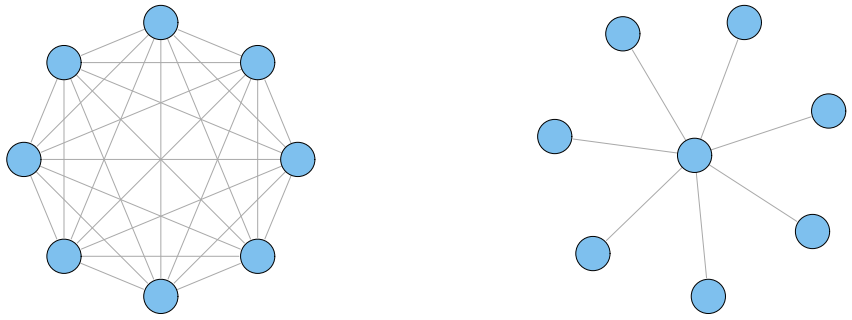
- Routing: how many hops does a message take to reach its destination?
- Search: How *easily* can an arbitrary object be located?
- Robustness: What is the degree of fault-tolerance of the network?
- Clustering: How many densely connected groups of nodes are in the network, and what are the sizes of these groups of nodes?

The rest of this introduction details the observations mentioned above, starting from an overview describing why network topology matters, going on to explain the nature and dynamics of network topologies, and finally to a description of the features of evolving network topologies. The problem statement, contributions, and outline of the thesis are presented.

1.1.1 Topology Matters!

The network topology has a direct impact on performance, resilience, and security of distributed applications. To illustrate the significance of network topology, consider the two canonical topologies, full shown in Figure 1.2(a) and *star* shown in Figure 1.2(b). Both topologies consist of 8 vertices but have very different topological properties, as can be seen in Table 1.1. The table's rows show the two considered network topologies and the columns show measures of three structural metrics, *average path length*, *edge connectivity*, and *average centrality*. Average path length counts the average number of hops between all pairs of nodes, edge connectivity measures the minimum number of edges that need to be removed to disconnect the network, and average centrality measure the number of times that a node appears in the shortest path between all pairs shortest paths.

In the case of the *fully connected* network, communication between any two active nodes is disrupted if all nodes fail concurrently, whereas in the case of *star*, communication is disrupted if the single central node fails. In other words *full* is more



(a) (b)

Figure 1.2: Example topologies: (a) fully connected and (b) star

resilient than *star*. On the other hand, the 7 links of *star* compared to the 28 of *full* make it more scalable, which might explain why *star* topologies, also known as hub and spoke architectures, are dominant on the Internet and in organizational networks. While the network topologies of regular structures such as the *full* and *star* topologies are well known and understood, the application-level properties of complex and irregular topologies such as the Internet remain an open problem that needs to be better understood.

Topology	**Average Path Length**	**Edge Connectivity**	**Average Centrality**
Star	1.75	1	2.62
Full	1	7	0

Table 1.1: Comparison of Global Properties of the *Star* and *Full* Topologies. *Edge Connectivity* refers to the minimum number of links that need to be removed to disconnect the network. Average Centrality measures the number of times a node appears in the shortest path between any pair of nodes.

1.1.2 The Nature and Dynamics of Network Topologies

Real-world networks evolve over time through dynamic nodes and links addition, removal, and rewiring. These dynamic events happen at the local level in the absence

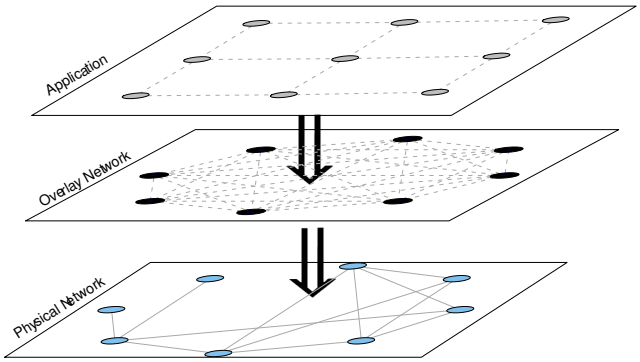


Figure 1.3: Application-Level Overlay mapped onto Physical Topology.

of centralized and global control, and have complex nonlinear implications at the application-level that need to be understood and managed.

In the last few years alone the Internet has seen exponential growth in the number of transactions, both at the network (e.g. BGP route updates and withdrawals) and at the application level (e.g. DNS, email) [6]. This rapid growth significantly impacts the performance, manageability and reliability of emerging networks. While the limitations induced by such a fast growth have been remedied, to a certain extent, through manual changes made to configuration tables (e.g. BGP tables), such ad hoc approaches are quickly becoming insufficient and impractical. While policies are defined by humans, their deployment, enforcement, and dependency conflict resolution, which is currently also human-centric, should be handled by autonomic mechanisms to optimize overall system performance. However, achieving this objective requires conceptual, physical and logistical modifications to existing systems and protocols.

A key issue that is significantly impacting emerging networks and applications, and that can be potentially addressed by autonomic techniques, is the absence of accurate knowledge of, and control over the actual topology of large networks. Network topologies define the link relationships between the nodes in the network and have a direct impact [7] on the security, resilience, and performance of a distributed system (see

Chapter 3). Large scale systems, such as the Internet, have grown out of minimizing cost and maximizing performance at the expense of flexibility [8, 9, 10]. In the case of the Internet, the criticality of these concerns have been highlighted in *Overcoming the Internet Impasse through Virtualization* [11] and *The Internet is Broken* [12]. While it is still possible to monitor the state, at the BGP level, of the approximately $65,000$ Autonomous Systems in existence to date [6], monitoring all the traffic inside these Autonomous Systems is a daunting and nearly intractable task. It requires days of data collection and processing to obtain a global picture of the network [13, 14], which is likely outdated by the time data collection is completed. It is for these reasons that in the past few years most distributed applications have been designed on top of or as *overlay networks*, giving network and application engineers more control over their target network.

While the physical network topology is hard-set and can not be changed using software alone, an overlay network can be rewired virtually in any desired way, thereby enabling dynamic software-driven configuration and management of the topology. Such overlay networks have been used to study problems of scalability, routing, resilience, fault tolerance, security and search in networked systems. The emergence of overlay networks as supporting platforms for the deployment of next-generation network applications makes the problem of topology control and selection a critical component of the network design problem.

The network topology can be virtualized according to a metric, e.g., geographic distance, bandwidth, or signal strength, that redefines the link relationship between the nodes. For example, in the virtual representation of a network as illustrated in Figure 1.3, the mapping between application-level requirements to the overlay representation and onto the physical network provides an abstraction that depends on the functionalities that each layer supports and provides. Furthermore, besides offering content

and context abstraction, virtualized topologies that are deemed beneficial to the network can be used to guide the deployment of physical connectivities.

In summary, next-generation virtualized networks further accentuate the problem of network topology selection and construction due to their ease of deployment and high dynamicity. An in-depth quantitative and qualitative study of various network topologies is presented in Chapter 3.

1.1.3 Local Perturbations Affect Global Properties

In the absence of global information, nodes gather local information that reflect partial views of the network. Modifications of the configuration of the network using nodes local information affect the properties of the global network.

The identification of appropriate metrics to analyze the network is a key requirement of the design and parameter space. In this thesis, a set of local and global metrics are presented. In particular, the local metric of network degree entropy as an information measure of neighborhood homogeneity is introduced. A novel mechanism of cluster detection based on network entropy is also presented. The number and size of each cluster is a global application-level metric that characterizes a network topology.

1.2 Problem Statement

The goal of the research presented in this thesis is to better the understanding of local adaptive strategies on networks application-level properties. To this end, we first identify network rewiring strategies based on local decisions and adaptations to evolve topologies, and study the emergent global properties of the resulting network.

The network topology reconfigurations consist of nodes making connections, disconnections, or rewirings at the local level. Inspired by the concepts of emergence and self-organization, this research tackles the following issues:

- The identification of canonical local structural metrics of a network.
- The evolution of network topologies of given degree distributions.
- The analysis of the emergent network structures that result from perturbations of the local metrics.
- The interpretation and correlation of local to global properties to better understand the impact of local network reconfigurations at the application-level.

The assumptions and definitions under which the problem of topology control is addressed are:

- In the limit of large networks, nodes may have limited view of the network.
- Nodes are assumed to be cooperative and not malicious.
- Nodes and Links can be added and/or removed dynamically.
- All edge weights are of constant unit cost.

The local and global metrics that will be used are summarized in Table 7.2 and are discussed in Chapter 2.

The proposed approach consists of evolving and adapting the network topology towards satisfying desired local structural properties. The approach is based on concepts of emergence, self-organization, optimization, and graph theory, and has three key aspects: first, the application-level properties of the network relating to path length, search, robustness, and clustering are determined. Second, each node computes local structural metrics. Adaptations are then performed in a decentralized manner where

local nodes apply neighbor selection policies, link addition, deletion, and rewiring, to evolve the topology. These network modifications are done using a simulated annealing optimization during which a move in the optimization process consists of exchanging two independent edges in the network. The move is accepted if it is optimizing or, if it is non-optimizing according to an annealing probability. Finally, the emergent properties of the resulting networks are correlated to the local perturbation strategies applied at each node.

1.3 Contributions

- A quantitative analysis of the impact of topology on network applications (Chapter 3).
- The introduction of a novel measure of network robustness (Chapter 3.
- The definition and evaluation of *degree network entropy* (Chapter 4).
- A survey, evaluation, and novel algorithm for cluster detection (Chapter 5).
- The demonstration of degree network entropy as an efficient local strategy to control average path length (Chapter 6).

1.4 Outline of the Thesis

The thesis is divided into three parts. The first part addresses *Topology Matters* and motivates the problem of topology selection. The second part covers the metrics of *Network Entropy, and Network Clustering*, addresses the importance and selection of respectively local and application-level structural properties in the evaluation and selection of a topology. Finally, the third part on *Topology Dynamics and Emergent Topologies*, presents mechanisms based on the selected network metrics and studies the properties of the resulting networks. More precisely:

- Chapter 2 covers background, related work, and a description of the tools used in this research. This chapter starts by showing how network topologies can be effectively represented and analyzed using graph theoretic formulations and metrics. Related research on network topology that use these graph metrics to approach problems in topology are presented. In particular, topology-aware methods use topological information at the node level to address local reconfiguration decisions. Topology modeling approaches attempt to find best-fitting statistical models of real-world networks by investigate correlations between graph metrics for the real and model networks. Emergent topologies combine topology awareness and topology modeling and study the bottom-up processes that explain the observed properties of an evolved topology. The software tools that were used to generate, evolve, optimize, and measure the network properties are introduced.

- Chapter 3 presents a quantitative and qualitative analysis of network topologies. It delves into the relationships between structure and function of various network topologies from regular to random. The chapter starts with the identification of a set of canonical, regular and non-regular network topologies, and goes on to present a quantitative analysis of the impact of these topologies on a set of network applications. Chapter 3 is central to the thesis as it motivates the importance of the network topology and its impact on application behavior and performance.

- Chapter 4 introduces the *degree network entropy* metric as a measure of information of a node's neighborhood degree homogeneity. Identifying relevant local metrics to assess structural and path information of a topology is an important problem because real-world networks are the result of limited information horizons. Furthermore, the decisions made at the local level yield global structures that impact application-level functionalities. This chapter analyzes the network entropy metric for a variety of topologies and a variety of evolving models of

topologies. It also shows how the network entropy metric can be used to quantify information for a given network topology.

- Chapter 5 defines the network clustering problem and presents a qualitative analysis of cluster detection for various network topologies. The number of network clusters in a network topology and the size of each cluster is a valuable topological property of a network. This chapter introduces a review of existing methods to detect clusters in a graph and presents a novel method of cluster detection based on network entropy.

- Chapter 6 presents an in-depth study of evolving network topologies with arbitrary structural properties using the previously identified local and global metrics. Two modeled topology instances, random and power law degree distributed, and two real-world networks, the Canadian Autonomous System and the Gnutella networks are considered. The evolved networks are evaluated with respect to the global emergent properties that result from the local adaptive strategies adopted by nodes locally. Correlations between the local metrics of transitivity, assortativity, entropy and application-level properties for routing, search, robustness, and clustering are addressed.

- Chapter 8 concludes the thesis and presents future work.

Chapter 2
Background, Related Work, and Tools

2.1 Introduction

The representation of a network topology as a graph provides a powerful abstraction to analyze the structural and flow properties of networks. This chapter first introduces a brief review of fundamentals of graph theory, from common graph representations to the metrics that will be extensively used in the rest of the thesis. This chapter goes on to addresses related work in the area of network topology awareness, network topology modeling, and self-organizing evolving topologies. The review is drawn from an extensive and active literature of which the research papers that have been most influential in shaping the work in this thesis are [5] [15] [16] [17] [18] [19]. The glossary in Appendix A provides a reference to the terminology used throughout the thesis.

2.2 Graph Theory Fundamentals

Graph Theory is a field of Mathematics that was officially born around 1736 with the now famous problem of *the seven bridges of Königsberg*. The problem was to find a route that crossed all bridges in the German town of Königsberg only once. Paths that solve such problem are now known as *Eulerian paths* after *Euler* who showed using a graphical representation of the bridges that the condition for such a path to exist was to have all endpoints on the path have an even number of links. Euler showed that it was not possible to find a path crossing all seven bridges without traversing a bridge more than once. Since then, the field of graph theory has extended theoretical Mathematics

and been applied to the formulation, analysis, and derivation of many scientific fields including Biology, Physics, Chemistry, Social Sciences, Computer Science and more.

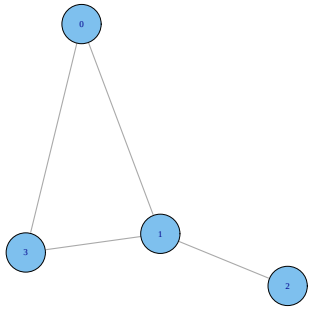
The strength of graph theory lies in the abstract representation of a problem into a set of nodes (or vertices) and links (or edges), in which two related nodes are connected by an edge. This relation applies equally well to molecular structures in which atoms are the nodes and edges the valency between atoms, or in social networks where nodes are individuals and edges represent relationships such as friendships or professional acquaintance between individuals.

2.2.1 Graph Representations

A Graph G is denoted as $G = (V, E)$, where V is the set of vertices, and E the set of edges. When relationships between nodes in the graph are not symmetric, the edges of the graph are considered *directed*, otherwise the edges are *undirected*. For example, the graph of the World Wide Web is directed with edges representing web links between pages as nodes. A link from one page does not imply that a back link exists from the page that is pointed to. If a graph contains more than one path to and from a node, it is said to be *cyclic*, otherwise it is referred to as *acyclic*. A tree graph or chain graph are examples of acyclic graphs. There are two common graph representations, matrix form and list form.

Matrix Representation: The matrix representation of a graph consists of a two-dimensional integer array indexed by all nodes in the Graph, say from 0 to N. Every matrix entry $M[i, j]; i, j \leq N$, contains an integer number that represents the relationship between node i and node j. In an adjacency matrix representation, the entry $M[i, j]$ is 1 if there is an edge between the two nodes, and 0 otherwise. In a Laplacian representation, the entry $M[i, j]$ is -1 if $i \neq j$ and there is an edge (i, j). For example, the sample graph in Figure 2.2.1 has the adjacency matrix representation $A_{i,j}$ and

Laplacian representation $L_{i,j}$:



$$A_{i,j} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$L_{i,j} = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

Figure 2.1: Sample graph with 4 nodes and 4 edges.

Once in matrix form, operations from Matrix and spectral graph theory [20] can be derived to formulate asymptotic analysis. However, this representation can be expensive in memory, and particularly inefficient when graphs consist of very large number of nodes with few edges between them. Such graphs are known as *Sparse Graphs* and benefit from an alternate representation of a graph, namely a list representation.

Edge List Representation: Using the list representation, every node in the graph *lists* the nodes that they are connected to. The difference with a matrix representation is that the nodes entries *ignore* the nodes that they are not connected to. If the graph is sparse, this representation results in a significantly more compact form. For example, the sample graph in Figure 2.2.1 has the following edge list representation:

$$E_{i,j} = \begin{pmatrix} 0 \to 1 \\ 0 \to 3 \\ 1 \to 2 \\ 1 \to 3 \end{pmatrix}$$

Graph Attributes: Nodes and Edges of a graph can be *augmented* with arbitrary information that is relevant to a problem or application. A classic example in network

flows considers the distance between two nodes to be an attribute of the edge connecting adjacent nodes. Such a representation is known as a *weighted* graph and is used to derive shortest paths and minimum spanning trees of a graph. Similarly, nodes can be augmented with characteristic values that reflect an arbitrary property of the node. For example, Figure 2.2 shows a compressed graph of the North American power-grid graph, the original graph contains 4941 nodes and 6594 edges, and is shown in the figure compressed to 40 nodes and 215 edges. The graph is the result of applying a *clustering detection* algorithm in which each cluster is represented by a single node and for which the size of a node is scaled in proportion to the size of the cluster. Clustering is discussed in detail in Chapter 5.

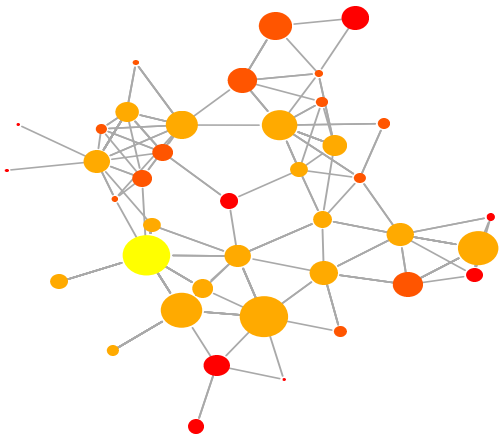


Figure 2.2: A clustered (*compressed*) representation of the North American electrical power grid.

2.3 Graph Metrics

The building blocks of a topology are *nodes* and *edges*[1], that assemble and form structures that are well or ill suited to fulfill certain application-level functionalities. Such *goodness of fit* can only be revealed based on observable graph metrics without which understanding the structures formed by topologies remains an art. Interpretations based on relevant graph metrics enable the evaluation of a topology and its *fit* to fulfill a certain function in a quantifiable way. The network and graph theory literature have addressed a set of metrics that measure local and global aspects of a topology [21, 22]. In this section we review some of the most common graph metrics and apply them to a set of topologies commonly encountered in networks. These metrics are now introduced and are summarized in Table 7.2 (see Appendix).

2.3.1 Fundamental Metrics

Number of Nodes and Edges: The relationship between the number of edges to number of nodes can determine feasible, i.e., connected, topologies from unfeasible, i.e., disconnected, ones. For example when considering N nodes and $N-1$ edges it is only possible to form a *connected network* by chaining nodes to each other in a linear fashion. This case is evidently the most straightforward and when more edges than nodes exist, the number of possible wiring of nodes immediately becomes combinatorial, therefore leading to known hard problems in topology such as the identification of a graph's automorphism class.

Degree: The degree of a node is its number of adjacent neighbors. It is the most fundamental structural property of a topology. The degree of a directed graph can relate to the *indegree*, i.e. number of incoming links, or *outdegree*, i.e. number of outgoing links, as distinct measures. Together, *In-* and *Out- degree* give the *total* degree. For an

[1]Throughout the paper *nodes* might interchangeably be referred to as *vertices*, and *edges* as *links*.

undirected graph the *In-* and *Out-* degrees are equivalent and are referred to simply as degree.

There is a direct relationship between degree and the number of edges, as the total number of edges in the topology corresponds to the sum of the degree of each node. For a directed graph of n nodes, the number of edges m is the sum over each node i outdegree d_i, $\sum_{i=1}^{N} d_i$. For an undirected graph it is the sum over all nodes i degrees divided by two, $\sum_{i=1}^{N} \frac{d_i}{2}$, as each edge is counted twice.

While degree is a local property of a node, once all nodes degrees are gathered, global information can be assessed and provides additional information of the network. Relevant degree-related metrics are *degree distribution* and *joint degree distribution*.

Degree Distribution: The degree distribution of nodes i with degree d_i written as

$$\frac{1}{n} \sum_{d_i} 1$$

gives the frequency at which a degree is represented in the network graph. The distribution is essential when fitting a network model to real-world network data. For example, the Internet router link distribution was believed (pre-1999) to follow a Poisson distribution because at the time the accepted model was that the Internet wiring, under no governing body, evolved as an Erdõs-Rényi random graph. Using network data collected from a sample of the Internet routers, several studies [10, 17] concurrently showed, *circa* 1999, that the Internet link distribution fit a power-law distribution, of the type $y = x^{-\alpha}$ with α typically between 2 and 3. Note that as a statistical measure, degree distribution is only meaningful when the sample size or number of nodes is large and isn't relevant for small networks, for example tens of nodes.

Joint Degree Distribution: Knowing the degree distribution is significantly more informative than degree alone and is extensively used to analyze graphs. However, an identical degree distribution can exist for many graphs with very different structural

properties. The Joint Degree Distribution provides additional information that quantifies the joint node degree to average neighborhood degree. The computation of the joint degree distribution is performed by averaging the neighborhood degree of every node. This information highlights structural properties of the network by relating to degree homogeneity, i.e. whether nodes connect to nodes with like degrees or unlike degrees. In the network literature this measure of similarity is also referred to as *assortativity*, when nodes tend to connect to like-degree nodes, and *disassortativity* when nodes connect to unlike-degree nodes. Practically, *assortativity* can be measured using Pearson's correlation on the degree of every node.

Transitivity: also commonly referred to as *clustering coefficient*, measures the probability that a node's neighbors are themselves neighbors. A high transitivity value is indicative of a cohesive network, where alternate paths to and from nodes are common within a node's neighborhood. There are currently two formulations for transitivity, one is expressed as a ratio between k, the number of edges that exist between neighbors, i.e., triangles, to the total possible number of edges that could exist between neighbors, also known as *triples*, expressed as

$$C_1 = \frac{k}{\frac{n(n-1)}{2}}$$

for undirected graphs and $\frac{k}{n(n-1)}$ for directed graphs. The other formulation takes the average transitivity over all local transitivity computed at every node as

$$C_i = \frac{number\ of\ triangles\ connected\ at\ vertex\ i}{number\ of\ triples\ centered\ at\ vertex\ i}$$

and the total transitivity $C_2 = \frac{1}{n}\sum_i C_i$.

Edge and Vertex Connectivity: Considering a connected topology, the edge connectivity is the minimum number of edges that need to be removed to disconnect the network. The edge connectivity of a connected acyclic graph is obviously 0 but can be

higher for a high *transitivity* graph, and reaches a maximum of $n-1$ for a fully connected network. The vertex connectivity is the minimum number of nodes that need to be removed to disconnect a graph.

2.3.2 Path-related Metrics

The previous metrics presented some fundamental structural graph metrics, more can be said about a network by measuring various properties obtained by *walking* along the paths between nodes. Path lengths are measured from source to destination, and can reflect either the shortest path and subsequently *minimum spanning tree*, or longest shortest path, i.e., *diameter*, or the number of shortest paths going through a node, i.e., *betweenness centrality*. These paths related metrics are now introduced:

Shortest Path: The shortest path is the path that leaves a source node to a destination node and traverses (hops) the fewest number of nodes. The general idea to compute a single source shortest path is to attempt to reach every other node in the network, and at every step maintain path information from every node to every other node, if an alternate path between two nodes can be achieved with fewer hops than is current, the alternate path becomes the new shortest path. This process is repeated until all nodes are explored and all paths are checked for being of minimal length.

Minimum Spanning Tree: A spanning tree is an acyclic structure that traverses every node in the network. A minimum spanning tree (MST) is a spanning tree on a weighted network, where edges between nodes are given attributes (i.e., weights) and the spanning tree is built such that the sum of all edge weights is minimized. There exists several known processes to construct a minimum spanning tree from global information, one is to select edges of minimal weight and add them to the spanning tree as long as there are no cycles and every node is traversed once, known as *Kruskal's algorithm* [23], the other is to walk from a source by selecting the edges of minimal

weight as the path to every other node goes along, in a manner similar to single source shortest path, known as *Prim's algorithm* [23]. Computing the MST is very important in many applications that require methods to reach all nodes in the network more efficiently than by *flooding* or broadcasting messages to the entire network.

Vertex Betweenness Centrality: The number of shortest paths that traverse a node is indicative of the *importance* or *centrality* of that node. This measure of centrality is very important in estimating the resilience of a network to attack or failure. The more paths traverse a given node, the more important that node is likely to be and therefore the more disrupted the network would be if that node is removed. Betweenness centrality is computed based on the all pairs shortest paths, and considers for each endpoints pair the ratio of the number of times a node appears on the shortest paths between all pairs of nodes.

Edge Betweenness Centrality: Similar to the Vertex betweenness centrality but is computed per edge.

Diameter: The longest shortest path is the diameter of the network. Besides giving the maximum number of hops necessary to go from and to any node, the diameter does not reflect any average or general structural property of the network. However, while a high diameter might be due to a single unusually long path in the network, a low diameter is indicative of a highly cohesive network.

Graph Spectrum: A graph is numerically represented either in *matrix* or *list* form. As previously mentioned, common graph matrix representations are the Adjacency and Laplacian forms. From an implementation and representation point of view, it is more efficient to represent large sparse graphs in list form rather than matrix form due to the ratio of edges to nodes. However, for smaller graphs or when possible, the

matrix representation of a graph enables the application of matrix theory, and in particular the identification of the eigenvalues and eigenvector sets corresponding to the graph. General properties and bounds can be derived from the graph using the eigenvector/eigenvalue sets, and most importantly the strength of the connectedness of the graph which is characterized by high eigenvalues [24].

Research efforts related to network topologies can be broadly classified as related to “topology awareness”, “topology modeling”, and “emergent topologies”, this classification and a summary of the contributions of each field is presented in Table 2.1. These research areas further develop into sub-areas or applications as illustrated in Figure 2.3. These three broad categories are discussed in more detail in the rest of this section.

Application class	**Description**	**Main Result**
Topology Awareness	Use local information to assess the current and next state of the system	Structured or geographic-based systems can theoretically achieve logarithmic-time data propagation.
Topology Modeling	Identify best-fitting statistical models to the observed data collected from real topologies	Large unstructured networks tend to follow a power-law distribution.
Emergent Topologies	Build on the concept that the whole topology has features that are greater than the sum of its parts	Solutions to known NP-complete problems to date are provided by heuristics that build on the emergent property (e.g. ant algorithms)

Table 2.1: Classification of research directions related to network topology

2.4 Topology-Awareness

Recent research in virtualization [11] is in favor of data access protocols that virtualize content distribution, abstracting the data from its physical location into its virtual network location. However, this communication paradigm has severe drawbacks for location-dependent applications such as:

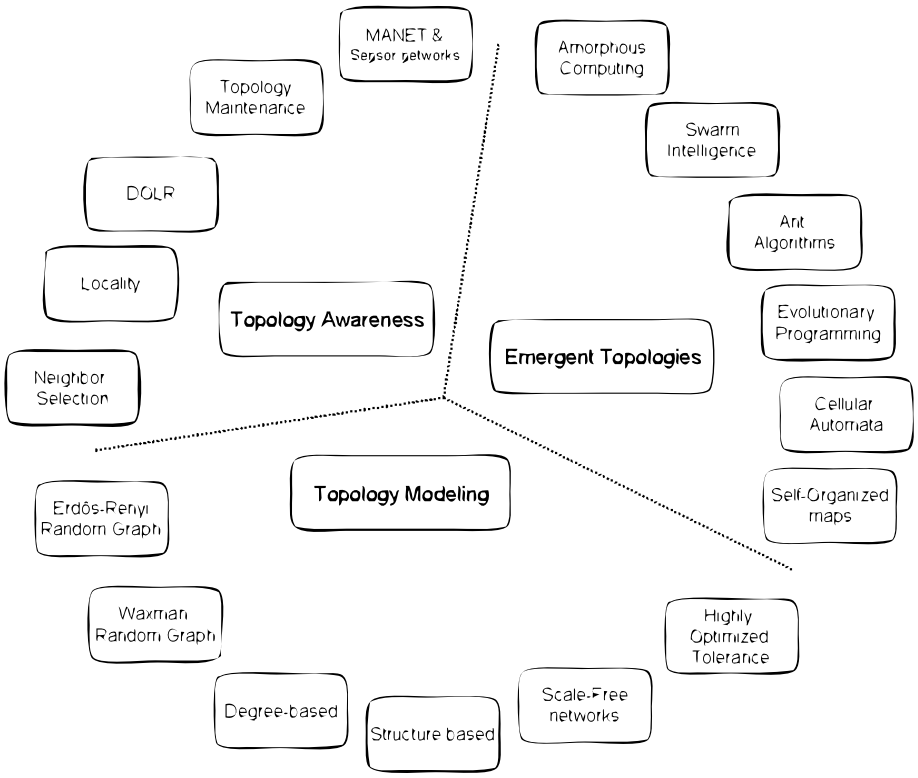


Figure 2.3: Scope of research related to network topology

- Ad-Hoc Routing: mobile networks that can communicate only within a specific geographical space.
- Security: confidential data that should be transmitted only within trusted networks.
- Data Management: data mining and information retrieval systems that collect and catalog information based on locality.

Such applications need to maintain a proximity view of their neighbors and, as a result, are termed *locality-aware* or *topology-aware*. The maintenance of a correct and persistent view of a dynamic system requires dedicated synchronized protocols that are either non-existent, or cannot be implemented over an unreliable messaging substrate. This makes finding the appropriate nodes to communicate with in an ad-hoc manner a difficult and challenging problem. Topology maintenance requires mechanisms to collect information from a set of nodes in the system in order to derive a global view.

The cost of maintaining a full view of the network is proportional to the size of the network [25]. While such maintenance protocols can achieve logarithmic time for infrequent changes, the number of messages exchanged grows exponentially for frequent changes. A more recent, but conceptually similar idea, is to use geographic locality for topology building. Several methods have been proposed to maintain geographic locality when a node joins a network. These include approaches based on landmark servers for position calculation [26], and on translating network distance into geographic distance [27]. However, as in the case of topology maintenance systems, these solutions do not scale well and in the case of landmark servers, require reliable and available nodes present in the network at all time in order to successfully retrieve the position of a node. Researchers at the University of Bologna have used topology awareness to evolve a network towards a desired topology [19]. Their work has demonstrated the efficiency of the gossip algorithm in reaching *eventual consistency* among nodes in a large network, and has shown that their algorithm can evolve a large network of nodes towards a given topology in a small number of message passing cycles. Other more recent techniques for locality and topology awareness, such as *Geographic Layout*, *Proximity Routing*, and *Proximity Neighbor Selection (PNS)* [28, 29] are evaluated in [30]. These geographic or proximity topology-aware protocols determine an optimal neighbor to forward data to and build structured and unstructured networks. However, these mechanisms of evolution and adaptation of the topology have been essentially focused on geographic locality and have not addressed the dynamic optimization of a network based on local and global structural metrics.

Topology awareness has also been used to achieve fault tolerant routing. For example, the Resilient Overlay Network [9] (RON) project addresses the problems raised by the BGP routing failures between Autonomous Systems. RON proposes an overlay network protocol to dynamically determine a new route for the packets to go around

a failed node. RON showed to successfully reroute data around a fault on small networks of 50 to 60 nodes, but did not address structural properties of the topology and their impact on application-level functionalities. Similar issues are also encountered in peer-to-peer networks, in which unreliable communications are established between peers in the absence of centralization. One such class of P2P applications, *Distributed Object Location and Routing* (DOLR) [31] uses *locality awareness*, *proximity routing*, *data replication*, and *soft-state maintenance* techniques to guarantee reliable and high performance search in the P2P system. Another area where topology-awareness is used is mobile ad-hoc networks (*MANET*). The cost of routing in MANETs largely influences the power consumption of the interacting nodes. Each node in a MANET maintains a local view of its neighboring nodes, i.e. a local map of the topology, and when optimized, can improve the performance of the application and reduce the power consumption of the nodes. One such optimization is used in Ascent [32], an energy-saving protocol for sensor networks. Ascent uses local topology information and the density of packet loss to determine the node's current and next state and uses Directed Diffusion [33] to build a global view of the topology.

While our research shares conceptual affinities with this work, particularly on the mechanisms for topology modifications, our focus is on understanding the impact of local reconfiguration strategies at the application-level.

2.5 Network Topology Modeling

In contrast to the approaches presented in Section 2.4 that grow a network using local knowledge of the topology, the network modeling field looks at real-world networks and attempts to identify the generating principles and best statistical fit between model networks and their real-world counterpart. The comparison of the networks is typically performed using new metrics that invalidate a prior model in favor of a new model for which the new metric is validated. For example, while the Internet topology was

originally believed to be a random graph [34] and assumed to have a Poisson degree distribution it was later shown to have a power-law distribution. Modeling of the Internet topology is actively researched and rendered difficult due to the rapid growth and frequent changes of the network. Five models of the Internet topology, random, Waxman, Doar-Leslie, Exponential, and Locality, were presented and compared in [35, 36], each model introduced a new metric that invalidated a previous model. The most current accepted model is derived from the Highly Optimized Tolerance model introduced in [8] that takes into consideration the economic and human costs as part of the network model and has been termed the “robust yet fragile” model.

2.5.1 Models of Complex Networks

The modeling of complex networks from biology, technology, and sociology have led to important discoveries towards understanding the generating principles of real-world networks. A “good” model enables the generation of networks with desired characteristics that are applied to study or simulate the behavior of the modeled large complex systems without depending on real-world data.

A complex network can be modeled with nodes joining a network in discrete time steps. At each step a node establishes connections to nodes already in the network. The number and target of these links are the two fundamental variables of the model. For example, the target nodes can be selected at random amongst all nodes, or preferentially, in which case a measure of preference is necessary to guide the connectivity process.

Models of evolving networks may also consider aging, a process by which links between two nodes are removed if a satisfying condition to maintain the link is not met. Aging enables the study of the dynamics of networks and is an essential characteristic of real-world networks. For example, in social networks, new links are formed through

the discovery of new acquaintances, or friends of friends, and dropped when affinities change.

The main models of evolutionary networks can be categorized as:

- *Random*: The existence of a link between any two nodes is drawn from an arbitrary probability distribution. The network may evolve with or without aging.
- *Preferential*: Links are determined based on a preference factor that measures an affinity between a considered pair of nodes. The network may evolve with or without aging.

These two types of models include as sub-categories: configuration model [37], Callaway Traits [38], and Small-World networks [39]. The next section briefly introduces each model.

Random Graph Models

Erdõs-Rényi Random Graph: The earliest random graph model was proposed by Rapoport and later reformulated by Erdõs and Rényi in a series of seminal papers [34] in the 1960's. This model established graph theory as an area of combinatorics and initiated the field of random graph theory.

In this original model, a graph with n nodes is constructed based either on a uniform probability, p, of existence of an edge between any pair of nodes and is denoted $G_{n,p}$ or based on a number, m, of edges in the graph and is denoted $G_{n,m}$. For the $G_{n,p}$ model, the average number of undirected edges in a random graph is $p\frac{n(n-1)}{2}$. This model grows a network with a Poisson degree distribution. All possible graphs constructed using these mechanisms belong to the class of Erdõs-Rényi random graphs.

Callaway Traits [38]: consider that each nodes in a network is assigned a type. A

type matrix determines the probability that nodes connect to each other. This network evolves in discrete time steps, at each step two nodes are chosen at random in the network, and are connected according to a predefined probability table based on the matrix of types.

Configuration Model: The configuration model [37, 40] is a class of random graphs that are built based on a degree distribution from which a degree sequence is chosen and all nodes degrees are paired to meet the degree sequence requirement. One process by which such an arbitrary degree sequence is constructed is to visualize each node as having "spokes" sticking out of it with no connection to any other node. The spokes of each node are then paired at random until there are no remaining nodes with unconnected spokes. The condition for such a network to be built is to have an even total number of spokes. Networks resulting from such a process might not be connected. If it is a requirement of the model to generate connected networks, the resulting networks have to be checked by running a connectivity check algorithm, such as breadth-first or depth-first search, on the generated network.

Watts-Strogatz Small-World Model: The model proposed by Watts and Strogatz in [41] considers a $2D$ lattice topology on which a fraction of nodes are rewired by creating long range connections outside of their neighborhood according to a set probability p. This randomized reconfiguration leads to what is better known as *Small-World networks* which have low transitivity (i.e., clustering coefficient) and low average path length. This model has been very influential to this research due to the relationships between the local reconfigurations and their impact at the global level.

Preferential Attachment Models

The preferential attachment model originated as a candidate model to explain the emergence of the power-law distribution exhibited by a variety of large-scale systems

in many sciences. Historically, power-laws have been found in income distribution (Pareto, 1897), city sizes (Zipf-Auerbach, 1913/1940s), word frequency (Zipf-Estouf, 1916/1940s), bibliometrics (Lotka, 1926), species and genera (Yule, 1924), economics/information theory (Mandelbrot, 1950s)[2].

Simon Model: Herbert Simon was the first to propose and document a preferential attachment model as a leading factor that exhibits power-law distribution in the context of Economy and wealth distribution [42]. The model is based on the observation that as new elements join an existing system, connections are formed with those elements already in the system that are most known or more popular, and is also known as the rich club connectivity, or "the rich get richer" phenomenon.

Price Model: In 1965, while studying citation networks, Derek deSolla Price found that the *in* and *out* degrees of coauthors followed a power-law distribution. His proposed model [43] to explain this observation was based on the concept preferential attachment similar to that expressed by Herbert Simon earlier.

Barabási-Albert Scale-Free Network: In 1999 several independently conducted studies [10, 17, 8] showed that the distribution of router links on the Internet exhibited a power-law distribution. Documents on the World Wide Web were also shown in [17] to be linked following a power-law distribution and termed *scale-free* networks according to the process of preferential attachment described by Barabási and Albert. More recent studies have elaborated on these initial observations, while revalidating the power-law distribution but giving different explanations for the underlying generating principles and manifestations of these properties [44]. A study of the evolution

[2]These historical references are copied here from a presentation titled "New Directions for Power-Law Research" by Michael Mitzenmacher.

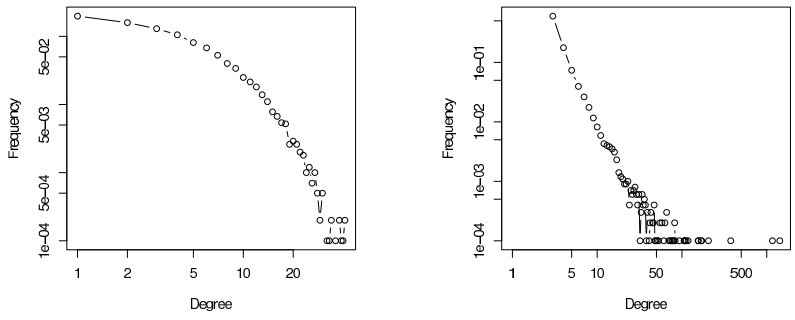
of the Internet topology design is presented in [44], in which a study of a single Internet Service Provider is used to extrapolate information about the global Internet and confirms the preferential attachment put forward in [17].

Summary: It is becoming key to many sciences to better understand the structural properties of complex networks, as well as to identify the appropriate metrics that are necessary to study and label such networks. Network topology modeling attempts to identify the guiding principles that drive the evolution of real-world networks. The models are compared to their real-world counterpart through specific metrics that act as *signatures* of a network. Network models that accurately predict the value of a measured property on the real-world and modeled network can then be used as generators of arbitrary networks that behave similarly to the real-world networks, and be used as platforms to simulate interactive and behavioral patterns.

2.5.2 Network Models: An Illustrative Example

Real-world networks evolve over time through dynamic nodes and links addition, removal, and rewiring. These dynamic events take place at the local level in the absence of centralized and global control, and have complex nonlinear implications at the application-level that need to be understood and managed.

For example, consider two simple discrete time evolving models, random wiring and preferential attachment. In the random wiring evolving model, at each step a node is added to the network and connects to a node already in the network with a uniform probability p that is a property of the model. In the preferential attachment model, at each step, a node is added to the network and connects to a node i already in the network with probability $P(i) = k_i^{\alpha} + \beta$, where k is the number of links of a node, α a preference exponent, and β the appeal to connect to an unconnected node. A probability P closer to 1 means that an added node favors nodes that have more links. The resulting distribution of number of links, shown in Figures 2.4(a) and 2.4(b), are



(a) (b)

Figure 2.4: (a) Link Distribution of Growing Random Graph with 10,000 nodes and 2 edges per step. (b) Link Distribution of Growing Preferential Attachment Network with 10,000 nodes and 2 edges per step, $\alpha = 1$.

significantly different for each process. The random model leading to an exponential distribution and the preferential attachment to a power-law distribution. Observations and lessons learned from the analysis of such dynamics is necessary to understand the evolution of large scale computer networks such as the Internet.

2.5.3 Statistical Modeling of Networks

Large-scale networks can be described as a large set of interconnected nodes that are theoretically best described through a stochastic formulation of the nodes, edges, degree distributions, or other arbitrary network properties. One explicit formulation that has been used and shown successful in recent years is that of generating functions [17, 37, 45] based on the degree distribution of a network. Using generating functions, a network with arbitrary degree distribution can be represented using the mathematical notation of power-series [45]. Other properties of the networks can be obtained by deriving the moment generating function of the degree generating function.

Probability Generating Functions

Consider the probability distribution of vertex degrees k, the generating function, $G_0(x)$ can be expressed as:

$$G_0(x) = \sum_{k=0} p_k x^k,$$

where p_k is the probability that a randomly chosen vertex on the graph has degree k. The distribution p_k is correctly normalized such that

$$G_0(1) = 1.$$

Probability generating functions have properties that make them particularly well suited to the study of evolving networks, especially regarding its *Derivatives*, *Moments*, and *Powers*.

Derivatives of the probability generating function of p_k is given by the k^{th} derivative of G_0 according to

$$p_k = \frac{1}{k!}\frac{d^k G_0}{dx^k}|_{x=0}.$$

Moments of the probability generating functions give the mean of the generating function given by the k^{th} derivative of the generating function. For example, the first moment, is given by the first derivative, and expresses the average degree z, i.e., average number of neighbors, of the nodes

$$z =< k >= \sum_k k p_k = G_0'(1).$$

Higher derivatives give higher moments, for example the variance, i.e., the second moment is given by

$$v =< k^2 >= G_0''(1)G_0'(1).$$

and in general, the k^{th} moment is given by

$$< k^n >= \sum_k k^n p_k = [(x\frac{d}{dx})^n G_0(x)]_{x=1}.$$

Powers of the generating function express the distribution of the total of a sum of independent realizations of an observed property of the network. For example, for m nodes chosen at random in a large network, the distribution of the sum of the degrees of those nodes is generated by $[G_0(x)]^m$.

Using this formulation, structural properties of large complex networks such as the mean component size, average number of neighbors, and average path length, can be formulated analytically and are derived in [37]. The mean component size $< s >$, is expressed as

$$< s >= \frac{1}{1 - z + zS}$$

where z is the average number of neighbors of a vertex and S is the size of the giant component. The average number z_m of m^{th}-nearest neighbors is

$$z_m = G_1'(1) z_{m-1}.$$

which further reduces to

$$z_m = [\frac{z_2}{z_1}]^{m-1} z_1.$$

Hence, the average number of m^{th} nearest neighbors can be determined based on the number of 1^{st} and 2^{nd} order neighbors alone. The typical length l of the shortest path between two randomly chosen vertices on the graph is analytically expressed as

$$l = \frac{ln(N/z_1)}{ln(z_2/z_1)} + 1.$$

As mentioned in [37], such "*result is only approximate for two reasons. First, the conditions used to derive it are only an approximation; the exact answer depends on the detailed structure of the graph*". In the face of such limited analytical formulation, these measures should be computed independently for each considered network.

2.6 Emergent and Bio-Inspired Approaches

Emergence, or the emergent property, is a characteristic of some complex systems by which the number of possible interactions between the elements of the system is so large that the system as a whole may appear greater than the sum of its parts. For example, the World Wide Web exhibits an emergent property, as links under no centralized control follow a power-law, rather than random, distribution. There are several research efforts that build on the concept of emergence to construct and manage network topologies [46, 19]. While our research is also based on emergence, it differs in the strategies used to adapt node connectivity and evolve the network topology, as well as the metrics used to evaluate candidate adaptations. Other related work include efforts inspired by self-organization [47], nature and biological systems [48] such as Amorphous Computing [49], Swarm Intelligence [50, 51], and Cellular Automata [52].

2.6.1 Amorphous Computing

Amorphous Computing applies concepts from biology and evolution to develop computer languages (Growing Point Language [53]) and decentralized evolving systems [54]. Each computing cell is viewed as equivalent to an organic cell that is guided by its environment to determine its next state. The cells follow virtual chemical gradient and density trails to identify their position and direction of evolution.

2.6.2 Swarm Intelligence

Swarm Intelligence is inspired by social and behavioral theories in the animal and human kingdom. For example, foraging, nest building, and burial activities of social insects (ants, wasps, termites) follow local rules applied by each entity with no knowledge of the whole. Similarly, flocks of birds, schools of fish, and herds of mammals follow local rules that lead to structures with emergent properties.

2.6.3 Cellular Automata

Cellular automata is a mathematical model that was introduced by Jon Von Neumann. It is characterized by the evolution of a set of nodes with deterministic neighboring rules that interact and form, in non trivial cases, unknown global behavior with or without apparent pattern formations. While some games, such as the rules of the game of life yield self-sustaining patterns that are now well understood, cellular automata rules result in complex non-linear and non-deterministic patterns that remain unexplained. In [18], an asynchronous cellular automata is evolve and shown, for specific considered rules to manifest a surprising degree of structure.

2.7 Summary

The study of the effect of linking strategies between nodes in a network is a complex and nonlinear problem that is combinatorial in nature. It has been mainly addressed in the theoretical sciences such as graph theory and statistical physics. In the applied sciences, structured and unstructured topologies have been studied in areas such as content management and topology-aware approaches. However, the problem of studying the impact of local rewiring strategies and their implication at the application-level has received little to no attention from the network engineering research. This problem requires urgent attention to better predict the evolution of next-generation complex network structures. Results from recent research show that:

- Large-scale man-made systems appear to follow a power-law degree distribution.
- Nodes in large scale networks can reach eventual consistency using a small number of message-passing cycles.
- In a structured overlay network, information can be published and retrieved in logarithmic time.

While these results are groundbreaking in the understanding of large-scale complex network, more questions remain unanswered and are addressed in this thesis: (1) Are there canonical metrics that quantify a network topology, both at the local (i.e., node) and global level? (2) Are network perturbations based on the local metrics correlated with the application-level properties? (3) Can local reconfiguration heuristics yield predictable application-level properties?

Further, the strategies proposed by emergent and Bio-inspired computing share similarities in the manifestation of global properties from the interaction between small and simple parts. As a result, these approaches are inherently non-deterministic and their quantification remains little understood. Existing research efforts in this area raise important questions about the evolution of networks, the distributions that are manifested, the quantification of their resilience, security, and efficiency, and the rules of evolution required to obtain a desired behavior.

This research seeks to understand the emergent application-level impact of perturbations of network topologies based on local optimizing structural properties alone. This is achieved by analyzing networks, both from evolving network models as well as real-world networks, and using topology-aware mechanisms to obtain information from the underlying topology at the node level to determine the next system-state. While this research relies and has been inspired by graph theory, topology-awareness, topology modeling, and self-organization and the emergent property, it is unique in the identification of local structural metrics, and application-level properties, and the investigation of correlations between the optimized networks based on local metrics at the application-level.

2.8 Description of the Tools Used

The formulation of network topology as a graph and the computer representation of the graph as adjacency matrices and edge lists was based on the *igraph* [55] software library.

igraph is a library written in C, with interfaces to the *R* [56] statistical language, as well as Python. *igraph* offers many features that make it a tool of choice to study and analyze large complex networks from a structural perspective. Some of the features of *igraph* that were most relevant and used in this research include:

- High-level functions for generating random and regular networks.
- Routines for manipulating large graphs by adding, removing, or reconfiguring edges.
- The definition of structural properties such as degree or centrality.
- The implementation of advanced force-based layout generation such as the *Kamada* and *Kawai*, or *Fruchterman* and *Rheingold* algorithms to facilitate the visualization of small to medium-sized graph.
- A clean and well documented API that make it easily extensible.

Furthermore, the *igraph/R* interface brings the power of the *R* language to perform statistical evaluation on the graphs and related metrics such as map-reduce operations, vector and matrix arithmetic, list operations, and plotting on top of all the essential statistical features such as correlation, variance, etc...

As an example consider the following code that is used to measure the degree correlation of a graph g:

```
correlation<-function(g,m="pearson"){
   el<-get.edgelist(g)
   d1<-degree(g,el[,1])
   d2<-degree(g,el[,2])
   if(sd(d1,d2)==0) return(1)
   co<-cor(d1,d2,method=m)
   return(co)
}
```

The *igraph* functions in this illustrative example are *get.edgelist* and *degree*. The *R* functions are *sd* and *cor*, respectively for standard deviation and correlation. The first line gets the graph representation as an edge list, i.e., as a from/to relationship, and stores the degree of each node in a corresponding relationship. The correlation is computed using a Pearson correlation moment function provided by the *R* library. All network visualizations and plots in this thesis were generated using *R-2.5.1* and *igraph 0.5*.

Chapter 3

A Qualitative Analysis of Network Topologies

3.1 Network Topologies

This section presents some commonly used network topologies, from basic regular topologies, such as ring, tree, star, and lattice, to more advanced regular topologies such as hypercube, chordal ring, and Kautz networks, and finally to non-regular topologies, such as power-law and Poisson degree distributed. The topologies are then evaluated with respect to their properties for routing and search, robustness, and security. The network graph representation of a topology can be considered as having directed or undirected edges [1]. Figure 3.1 illustrates a high-level view of the range of topologies based on a three dimensional space that considers *modularity*, *transitivity*, and *regularity* as properties axis.

3.1.1 Basic Regular Topologies

Fully Connected: The all-to-all pairing of nodes in a network of N nodes requires $\frac{N(N-1)}{2}$ undirected edges.

Ring: The Ring topology has a uniform degree of two links per node. For a network of N nodes it contains N edges.

Tree: A Tree is an acyclic structure for which all nodes but the leaf nodes have two

[1] In this paper, all topologies, except the Kautz network, are considered undirected

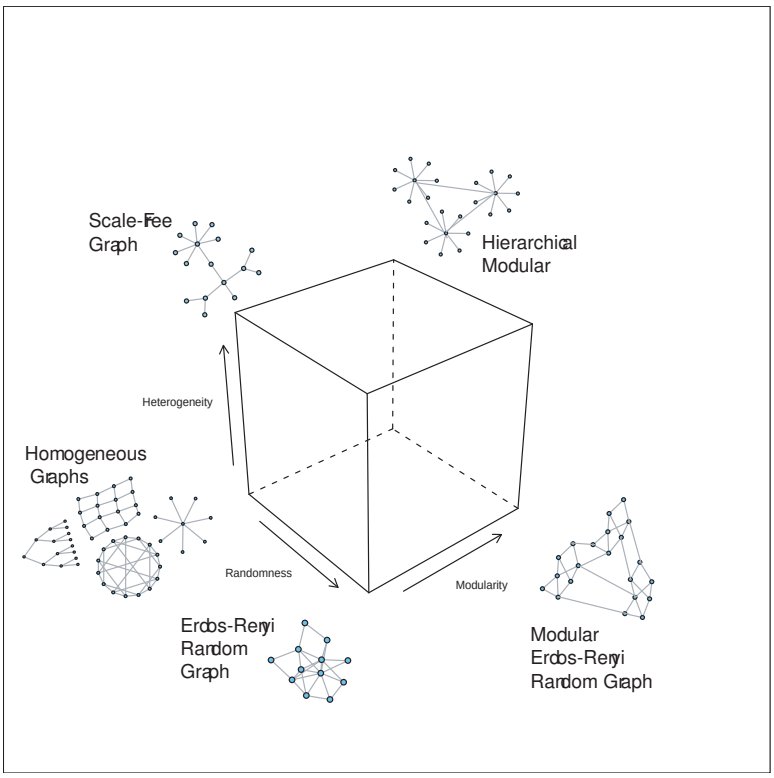


Figure 3.1: Network Topologies classified based on *Regularity*, *Transitivity*, and *Modularity*

edges. For a network of N nodes it contains $N - 1$ edges.

Star: The star topology has a central node to which all other $N - 1$ nodes connect to. This topology is very common in local area networks. It implements the hub-and-spoke architecture which is a common design pattern and consists of a single node that is common to all nodes in the network, that is used to route information to and from every pair of nodes. For a network of N nodes, the star topology contains $N - 1$ edges.

Lattice (Meshes and Tori): A mesh and a torus are related structures that differ in that tori are circular whereas meshes aren't. Circularity means that the extremities of the topology connect to nodes at the other extremity of the topology. For example, in a two dimensional mesh, all nodes but the ones at the corners have four neighbors, the four corner nodes have two neighbors, therefore, the total number of edges for a mesh

of N nodes is $2N - 2 * sqrt(N)$. In the case of a two dimensional torus of N nodes, the number of edges is $2N$.

3.1.2 Advanced Regular Topologies

Hypercube: Hypercubes are high-dimensional lattice topologies and are commonly used for embedding computations in parallel applications. An *n-cube* has 2^n nodes and $n2^{n-1}$ edges.

Kautz Network: A Kautz network of degree M and dimension $d + 1$ has $(M + 1)M^d$ vertices and $(M + 1)M^{d+1}$ edges. Kautz networks are well suited to fault tolerant topologies of parallel computer interconnects because they have the smallest diameter of any directed graph with N vertices and degree M.

Chordal Ring: An extension of a ring topology in which nodes are interconnected by chords going across the ring in a periodic manner. The number of neighbors in an N node chordal ring is between 3 and $N - 1$ (fully connected). When choosing the periodicity of the neighbors in an efficient manner, the chordal ring topology offers lower diameter and greater resilience than the ring topology.

3.1.3 Non-regular Topologies

The previous topologies are regular structures in that all nodes have similar degree characteristics. Non regular topologies have nodes with different properties that can form complex networks with arbitrary structural properties. We address two categories of such random networks, uniform random and preferential attachment.

Uniformly Random Networks: In a random network, each possible pair of nodes is interconnected with a certain probability. When the probability is drawn out of a

uniform distribution (i.e., for N nodes is $p = \frac{1}{N}$ for each pair), the resulting degree distribution is normal and centered around a mean of $p * \frac{N(N-1)}{2}$. This type of uniform random graph was proposed in [34]. This topology is particularly valuable when other topologies are compared to it to identify distinctive *unexpected* properties of the network.

Preferential Attachment Networks: Are a type of random graphs that *favor* certain nodes, hence giving a greater probability of attachment to some nodes and less to others based on preference. The generating principle of a preferentially attached network is that nodes are added in discrete time steps such that at every step the newly added node preferentially attaches to a node in the network that has greater degree. Such networks have been shown to exhibit power-law degree distribution due to the nature of the reinforcement by the preferential attachment to favor highly connected nodes in the network. The power-law distributed network has a degree distribution that follows a power-law of the type $y = x^{-\alpha}$ where α has been shown to be between 2 and 3 for various biological, technological, and social networks. This degree relationship expresses the fact that the node with highest degree is exponentially more connected than the second highest degree node and so on. This degree relationship has also been referred to as scale-free because of the self-similar relationships between nodes at various levels of degree connectivity.

3.1.4 Edge and Degree Summary

Table 3.1.4 presents a summary of the topologies with respect to fundamental properties of each topology, the number of edges and degree. Each topology considered has N nodes.

Topology	**Number of Edges**	**[Min-Mean-Max] Degree**
Full	$\frac{N(N-1)}{2}$	$N-1, N-1, N-1$
Ring	N	2, 2, 2
Star	$N-1$	1, $\frac{2N-2}{N}$, $N-1$
Tree	$N-1$	1, ?, 2
Mesh	$2(N-\sqrt{N}$	2, $\frac{4(N-2)}{N}$, 4
Torus	$2N$	4, 4, 4
Hypercube	$\frac{N(logN)}{2}$	d, d, d
Chordal Ring	kN	k, k, k
Kautz	$(M+1)M^{d+1}$	M, M, M
Random	$\frac{N^2p}{2}$	$1, Np, N$
Power-Law	$Nm-1$	$1, ?, N-1$

Table 3.1: Basic Topology Metrics. N is the number of nodes. d is the dimension of the Hypercube. p is the probability of two nodes being connected by an edge in the Random topology. k is the number of extra edges for each node in the chordal ring. M is the degree and d the dimension in the Kautz network. m is the preferential attachment exponent.

3.2 Network Applications

The network topology is the supporting structure on which a network application is deployed. It is the fundamental substrate on top of which communication between nodes takes place and higher order interactions are executed. The objective of this chapter is to show that the structure of the topology significantly affects the performance of an application. To this end, we consider the following *canonical* distributed applications, *Routing*, *Search*, and *Robustness*. In a first part, we briefly define these applications and then present an evaluation of the impact of topology on application performance.

3.2.1 Routing

Routing is the process of identifying a route from source to destination in a network of nodes capable of forwarding information as messages from hop to hop. Routing is an essential requirement of any message passing distributed system. The end to end delivery of a message involves finding a path between the endpoints, preferably shortest, but might also include finding alternate paths in case a link or node fails along the route. The most important decision in routing is to choose the node to forward information to such that the destination node is closer. The possible nodes to choose from are the neighbors of the node in the topology. While there are far too many existing approaches to routing to present in this section, the majority of approaches fall in one of three categories, *Distance-Vector*, *Link-State*, and *Ad-hoc*.

Distance-Vector Routing: In Distance vector routing approaches, nodes compile a vector of reachable nodes that is shared to all neighbors, every node then uses the exchanged vectors and computes the shortest paths to other nodes. Implementations of distance vector routing are provided by the Bellman-Ford algorithm, which works on a weighted graph that can contain negative edge weights. The algorithm is used for the Routing Information Protocol, RIP, but has the disadvantage of not scaling well, and

not reflecting changes in the topology quickly enough.

Link-State Routing: In Link-state routing every node *floods* the network with information about its neighbors. Once routers *hear* everything about the network they can calculate the best path to any host on any destination network. This can be done using *Dijkstra*'s shortest path algorithm [23], a variation of Bellman-Ford's algorithm for weighted networks with no negative edge weights. The most classical implementation of link-state routing algorithm is the *Open Shortest Path First* protocol.

Ad-Hoc Routing: There are two main categories of ad-hoc routing, *table-driven*, also known as *proactive*, and *on-demand*, also known as *reactive*. In table-driven ad-hoc routing every node maintains a routing table, when the topology changes, nodes propagate update messages to the network in order to maintain a consistent view of the topology. Examples of this approach are *Destination Sequence Distance Vector*, *Distance Vector Routing Protocol*, *Wireless Routing Protocol*, *Global State Routing*, *Hierarchical State Routing*. The differences between these systems are in the way the information is updated. In *On-Demand* routing routes are discovered as needed, the path remains valid until the route is not needed anymore or the timestamp for the route expires. Examples of this approach are *Ad-hoc On Demand Distance Vector Routing*, *Cluster Based Routing*, *Dynamic Source Routing*, *Temporally Ordered Routing*.

3.2.2 Search

Search is the process of looking from a source node, for a node or collection of nodes that match a query. The query is arbitrary and could be for content or resource. In distributed search the most challenging task is to obtain guarantees on the search, meaning that, with certainty, all items matching a query are returned to the requester. The network that is searched can be structured or unstructured. Structured networks offer the advantage of bounding the search time, possibly achieving logarithmic number of hops

to discovery.

In regular topologies, all nodes have an identical number of neighbors. Searching for an object in such networks is bounded by the average number of hops a message takes before finding the desired object. This expected search time (measured in number of hops) goes as $log_k(n)$ where k is the fixed regular number of neighbors of a node.

However nodes in the network could be structured to optimize for a given set of parameters such as geographic locality or bandwidth. Just as in routing, the network topology is essential in determining which of a node's neighbors is most likely to lead towards the desired content. A structured topology might offer similar bounds on the search process from every node, while an unstructured topology might have a high variance on the search bounds but offers more flexibility.

In contrast, irregular topologies evolve under no control and can therefore be locally and globally heterogeneous. Such heterogeneity renders the problem of discovery and searching in these networks difficult, often requiring a traversal of all nodes to guarantee discovery. Irregular topologies have been modeled using Newman's formalism [37] of probability generating functions. The probability generating function of a measurable property is expressed as a power series, $G_0(z) = \sum_{k=0}^{\infty} p_k z^k$ where p_k is the probability of a node to have degree k and z^k is the polynomial factor associated to a degree k. While probability generating functions are useful at setting some asymptotic bounds on some graph properties, their generalization remains challenging as the assumptions that are made to obtain the bounds render the problem more abstract and further from real networks. Such difficulties combined with graphs combinatorial explosion for many problems in search and optimization currently limit the applicability of analytical tools to real world networks, that are best assessed using experimental evaluation.

3.2.3 Robustness

The robustness of a network reflects its capacity to maintain functionality in the presence of changes or disruptions. One way to assess the robustness of a network is by measuring its edge connectivity, i.e., the minimum number of edges that need to be removed to disconnect the network.

However, the measure of edge connectivity does not quantify the *importance* of the edge that is removed in the network. For example, a fully connected network with 1 node at the edge of the network connected to the dense network will result in an edge connectivity of 1, but fail to recognize that this edge is not the most likely traveled and therefore not the most important to consider. To address this issue, we introduce a novel measure of robustness that uses the *edge betweenness* measure presented in Section 2.3, and is computed as shown in Algorithm 1.

Algorithm 1 A Measure of Robustness based on Betweenness Centrality

```
while G is connected do
  for all edges ∈ G do
    compute edge betweenness
  end for
  S ← sort edge betweenness in decreasing order
  remove edge = max(S) from G
  increment number of edges removed
end while
return number.of.edges.removed
```

This normalized measure of robustness has the advantage to reflect the centrality of an edge in the network, and returns a value closer to 1 when the total edges have to be removed.

3.2.4 Security and Cooperation

Identifying malicious activity in a distributed network can be addressed through cooperative strategies by which a set of nodes recognize a node as behaving in an anomalous

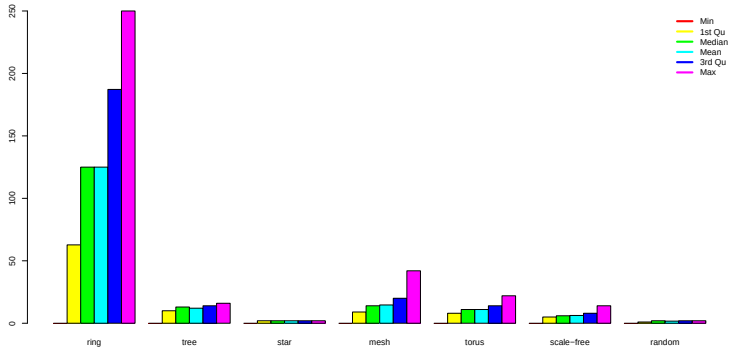


Figure 3.2: Shortest Paths for 500 Node networks

manner and flag it appropriately. Such a quorum forming is only possible if a set of nodes are in each others neighbor set. The transitivity metric presented earlier is important in that a higher transitivity reflects a cohesive network in which neighbors are themselves neighbors of each other, and can therefore vote upon a node's behavior or activity.

3.3 Evaluation

Using the metrics presented in Section 2.3, we evaluate the various topologies with respect to the applications presented in Section 3.2 and discuss the advantages and disadvantages of each topology for each metric. In this study, the weight distribution for all topologies is considered constant with cost 1. The evaluations with variable weight cost distributions is left as future work in this research.

3.3.1 Routing through Shortest Paths

The bar plot in Figure 3.2 shows a statistical summary for the distribution of the shortest paths for the network topologies presented in Section 3.1, each comprising of 500 nodes. The plot shows the minimum, first quartile, median, mean, third quartile, and maximum value for a shortest path in the respective topologies. The topologies will be

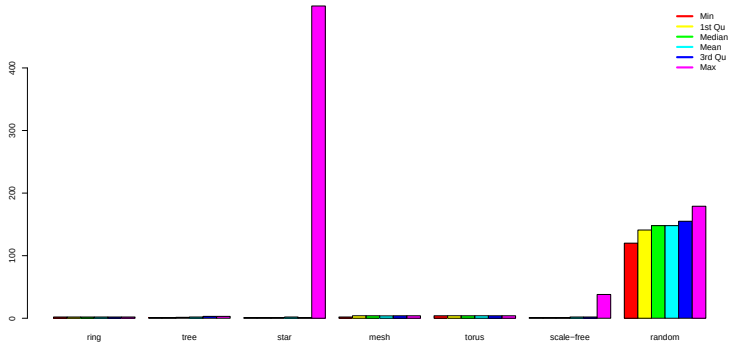


Figure 3.3: Degree of 500 Node network

referred to as *slow* when they manifest high average path length and *fast* otherwise.

The ring clearly appears as a *slow* topology with the highest average shortest path length. The star and random topology are the *fastest*, although this is at the expense of either high number of edges (for random networks), or high betweenness centrality (for star networks), as will be discussed below.

The bar plot in Figure 3.3 shows the same topologies and summary statistics for the nodes degree count. The star topology has a very high maximal degree, corresponding to the central node, and indicates that while on average nodes have very low degree, very few nodes have a very high degree (in the case of star just one). The random topology also stands out from the regular topologies as having nodes with higher average degree, indicating that the nodes have more edges than other regular structures. The scale-free network presents a trait similar to the star network, although of lesser scale and indicates that very few nodes have much higher degree than the average, hinting at the power-law nature of the degree distribution.

The shortest paths and the degree plots illustrated aspects of the topology relating to the distribution of degrees but fail to indicate if nodes in the network are more important (from a routing perspective) than others. This is shown in the bar plot in Figure 3.4, which shows the betweenness centrality for the same network topologies.

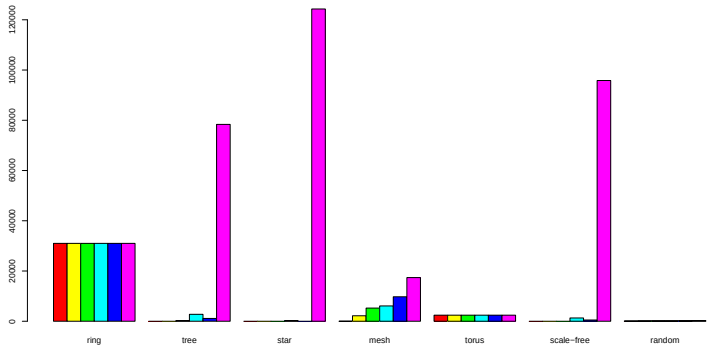


Figure 3.4: Betweenness Centrality of 500 Node networks

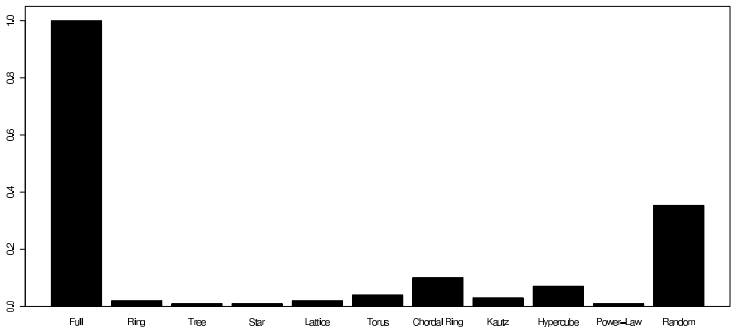
A high betweenness centrality indicates that some nodes are central to the network and used in routes to and from many nodes. This is strikingly visible for the star, the scale-free, and the tree networks. In a ring and random networks however, the distribution of routing responsibilities are equally distributed amongst all nodes, in particular the random network indicates the absence of betweenness centrality.

3.3.2 Robustness

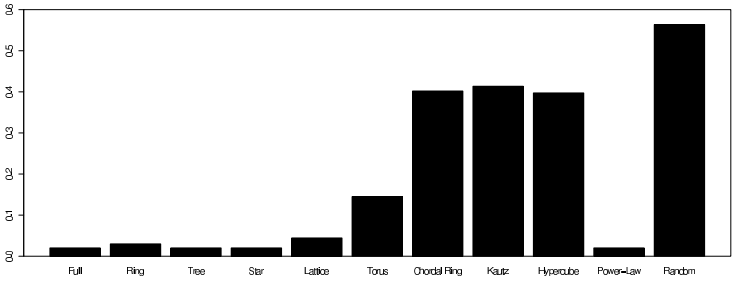
The bar plots in Figures 3.5(a) show the edge connectivity and measure of robustness of the 500 nodes network topologies considered.

The random network has the second highest, following the fully connected network, edge connectivity than all other topologies. This could be deduced from Figures 3.3 and 3.4, as the degree summary of the random network indicated that nodes are on average highly connected, and the betweenness centrality of all nodes remains very low. It is therefore not surprising that the random network would be highly resilient to link failure. The edge connectivity of the hypercube, chordal ring, and torus networks follow.

Figure 3.5(b) shows measurement of the network robustness using the metric introduced in 2.3. The removal of the most central edges in the network reveals the random



(a)



(b)

Figure 3.5: (a) Edge Connectivity and (b) robustness measures.

network as the most robust topology, followed by the Kautz network, the chordal ring, and the hypercube.

3.3.3 Search and Network Coverage

The following results show the network coverage ratio of each of the topologies presented in Section 3.1. The coverage is measured for each topology with 1000 nodes. The results show the average ratio of nodes reached per hop in the topology over all nodes. The maximum number of steps corresponds to the diameter of the network. This can be expressed as:

$$coverage_i = \frac{1}{N} \sum_{N} \left[\sum_{j=1}^{i} \nu_j \right]$$

, where N is the number of nodes in the network, i is the number of steps and can take values between 1 and D where D is the diameter of the network, ν is the number of neighbors at step j from the source node.

The nodes closer to the center of the lattice topology have better coverage than nodes closer to the sides. Figure 3.6 shows that it takes an average of 30% of the diameter to reach 50% of the nodes in the topology.

The advantage of the lattice is that it maps easily to geographical grids and fits well models of local information exchange while retaining a low average degree. However, the main disadvantage is that nodes at the extremities require larger number of hops to communicate, which might add significant communication load on nodes in the center of the topology when routing through shortest paths.

The tree topology forms a hierarchical structure that has a single root and a single path from every pair of nodes. Therefore the network coverage is greater for nodes closer to the root than nodes closer to the leaves of the tree. The root node of any subtree is also responsible for routing between its subtrees, therefore adding a larger required bandwidth to higher level nodes. This bandwidth issue was addressed in a modified tree structure known as *Fat Tree* which contains more links at every level

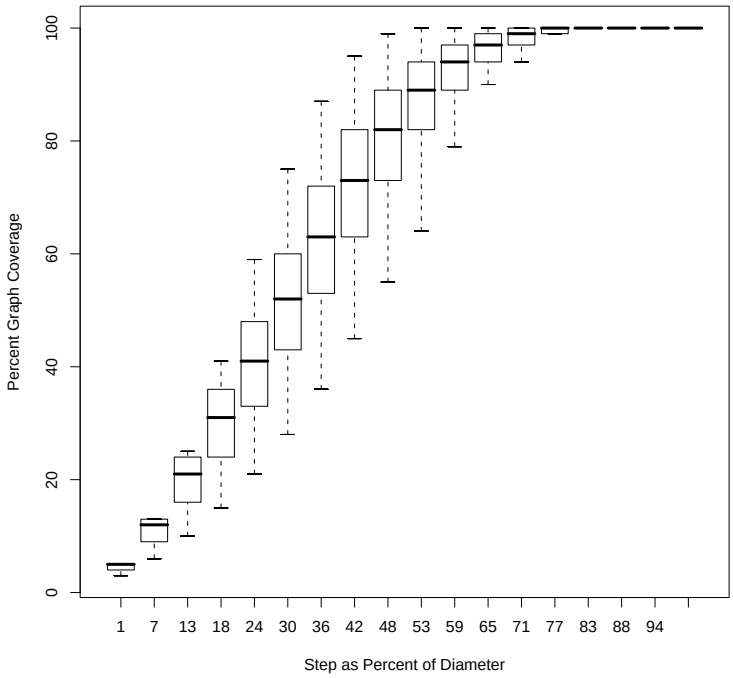


Figure 3.6: Average Network Coverage over a Lattice topology

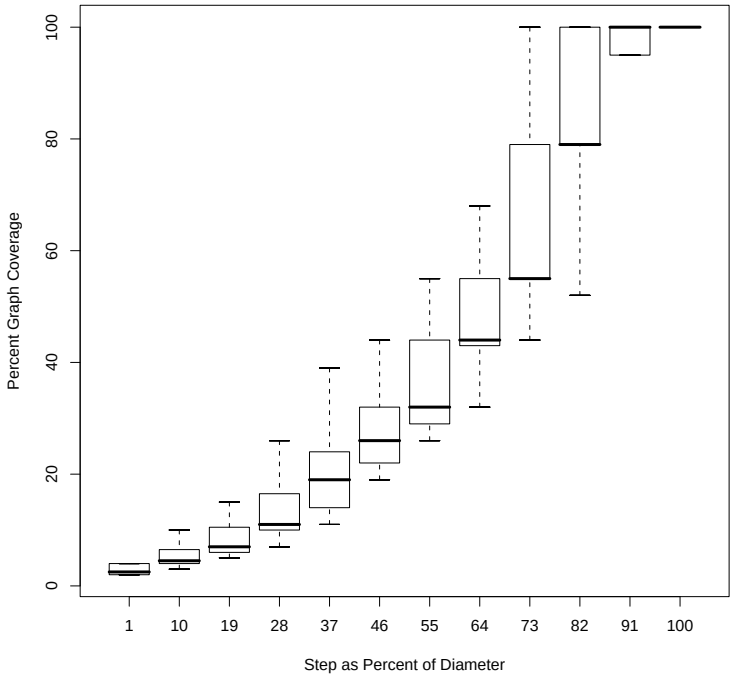


Figure 3.7: Average Reachability per node per step for a Tree topology

from the leaf to the root. The regular tree is not a resilient structure as the removal of a single link disconnects the topology, further, if a root link is disconnected the network becomes partitioned in half for a balanced tree structure.

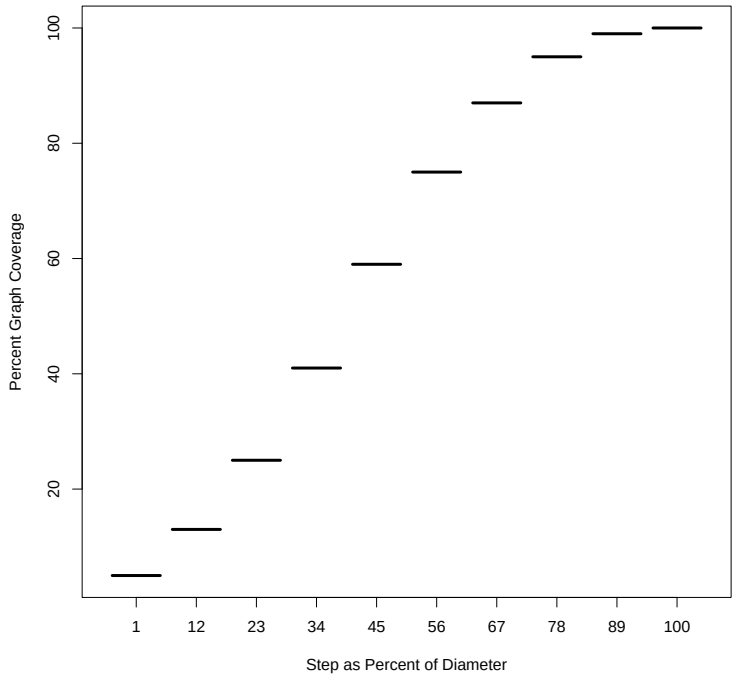


Figure 3.8: Average Reachability per node per step for a Torus topology

The torus, or *2D mesh with wraparound* as it is sometimes referred to, is a mesh topology in which nodes at the extremities are connected to each other. Therefore the node degree is perfectly uniform and it overcomes the issue of the mesh topology regarding corner nodes that require many hops in order to communicate. The network coverage in a torus is therefore much faster than in the mesh.

The Hypercube or *n-cube* as it is also referred to, has one extra edge for each dimension, further, the $(n+1)^{th}$ cube has twice as many nodes as the n^{th} cube. Higher dimension hypercubes offer higher degree of resilience as there are as many alternate paths as there are added edges.

The Kautz network is often applied for fault tolerant interconnect topologies of

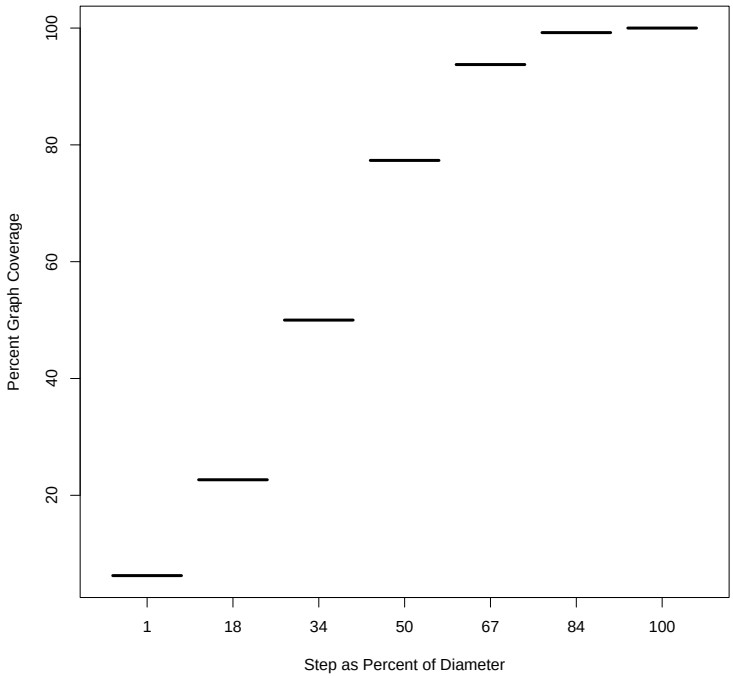


Figure 3.9: Average Reachability per node per step for a Hypercube topology

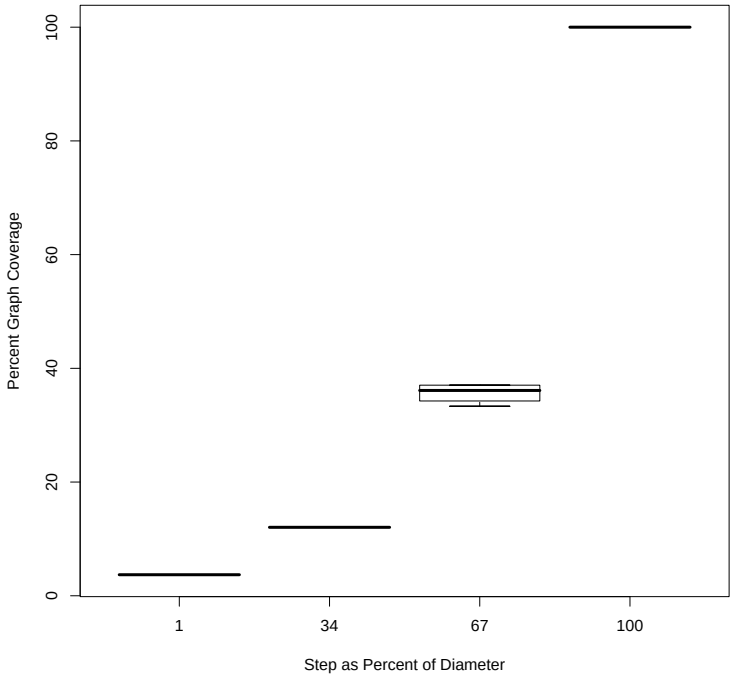


Figure 3.10: Average Reachability per node per step for a Kautz network

parallel machines. The advantages of the Kautz network is that it is efficient for embedding a high number of nodes with low diameter and low degree.

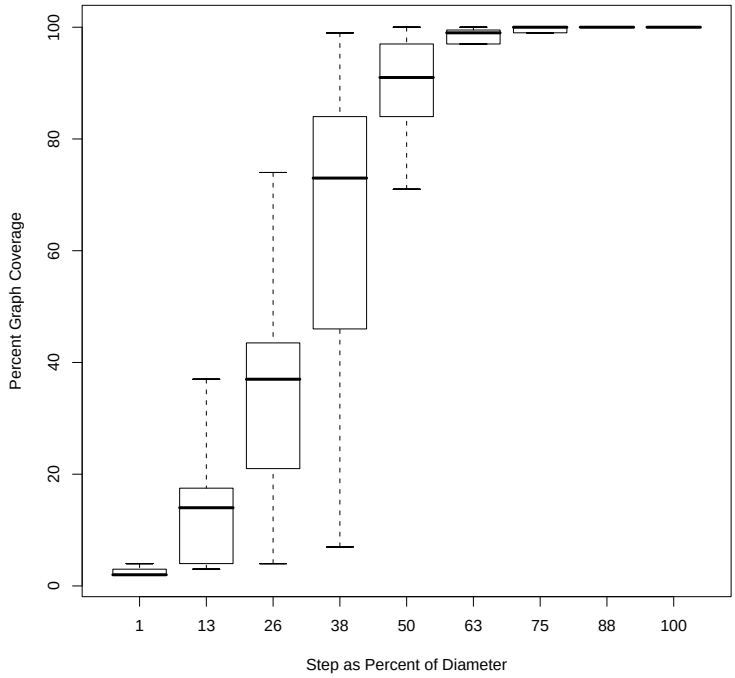


Figure 3.11: Average Reachability per node per step for a Scale-Free network

The non-regular scale-free or power-law distributed network has more nodes that are little connected than few nodes highly connected. In a hierarchical design in which some nodes are more important than others, such as the fat tree, this is a naturally occurring structure. It also emerges in many contexts other than computer engineering such as biological, social, and physics networks.

Besides the fully connected topology, the random network has a very high number of edges distributed randomly. This makes it very resilient to edge failure and gives the network a very low diameter.

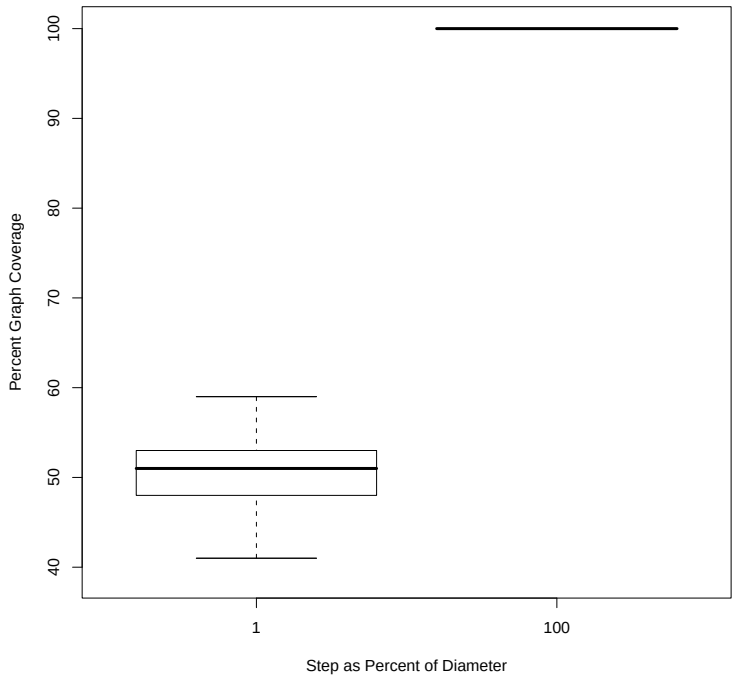


Figure 3.12: Average Reachability per node per step for a Poisson (p=0.5) network

3.3.4 Trust and Security

Figure 3.13 shows a statistical summary of the local transitivity for each topology under study.

It appears that most regular and advanced regular topologies have 0 transitivity. This is due to the fact that these regular structures do not contain any inter-neighbor edges. The fully connected network, random, and chordal ring topologies are the only topologies with a positive transitivity. The maximum transitivity is for the fully connected network at the cost of the maximum number of edges. Such a high transitivity makes identifying malicious activity easier because a voting round can include a maximum number of nodes in the network, however the cost of voting increases as the number of participants increases. So in such a case, the trade-off between number of participants and voting time can be detrimental to the optimal functioning of an application. The chordal ring being between a ring and a fully-connected network offers

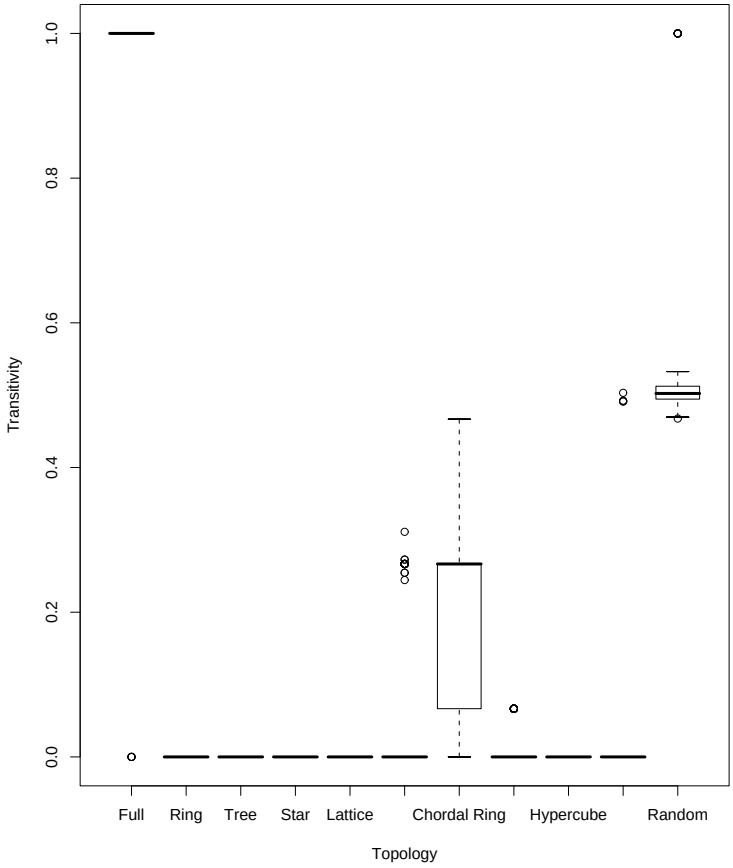


Figure 3.13: Transitivity of different Topologies

higher transitivity with yet fewer number of edges making it a very attractive candidate topology to identify malicious or dysfunctional nodes.

3.4 Exploring the Network Topology Design Space

We characterize the topologies with respect to the four fundamental properties, *number of edges*, *resilience*, *transitivity*, and *average path length*. We argue that an *optimal* topology is one that minimizes the average path length and number of edges while maximizing the transitivity and resilience. The problem can therefore be stated as an optimization problem in which the objective is to minimize the average path length and

the number of edges while maximizing resilience and transitivity:

$$maximize\{ \quad (\quad \tau * transitivity + \rho_1 * resilience + \rho_2 * robustness) - \\ (\quad \alpha * average.path.length + \epsilon * num.edges)\}.$$

With these design choices, it appears that while the fully connected network has optimal inter-node distance of 1 hop and presents very high resilience, it is very costly. The star topology also offers 1 hop to and from every node and has minimal number of edges, but has very low resilience. The ring topology has low resilience, low number of edges, and high diameter. The chordal ring and Kautz network appear as regular topologies of choice, with high resilience, moderate number of edges, low diameter and low average inter-node distances.

The optimal topology that satisfies a set of objectives can be obtained by varying the parameters $\tau, \rho_1, \rho_2, \alpha, \epsilon$. Figure 3.14 shows the results obtained when enumerating over all possible combinations of the four controlling variables between 0 and 2. The bar plot in Figure 3.14 shows the topologies on the horizontal axis and the number of times that each topology is selected as optimal out of the total 5^3 enumerations.

3.4.1 Interpretation of the Results

Figure 3.14 shows that the majority of the results favor the random and *fully connected* networks, the results that yield a different topology are shown in Table 3.4.1 through Table 3.4.1.

These results reveal that:

- The Kautz network appears to be optimal when the transitivity and edge connectivity resilience are not factored in the requirements of a topology.
- The Tree topology is optimal when the average path length and number of edges

Chordal Ring				
τ	ρ_1	ρ_2	α	ϵ
0	1	1	2	0
0	1	1	2	1
0	1	1	2	2
1	0	0	1	0
1	0	0	1	1
1	0	0	1	2
1	0	0	2	0
1	0	0	2	1
1	0	0	2	2
1	0	1	2	0
1	0	1	2	1
1	0	1	2	2
1	1	0	2	0
1	1	0	2	1
1	1	0	2	2
2	0	0	2	0
2	0	0	2	1
2	0	0	2	2

Table 3.2: Costs favoring the Chordal Ring Topology.

Kautz Network				
τ	ρ_1	ρ_2	α	ϵ
0	0	1	1	0
0	0	1	1	1
0	0	1	1	2
0	0	1	2	0
0	0	1	2	1
0	0	1	2	2
0	0	2	2	0
0	0	2	2	1
0	0	2	2	2

Table 3.3: Costs favoring the Kautz Network Topology.

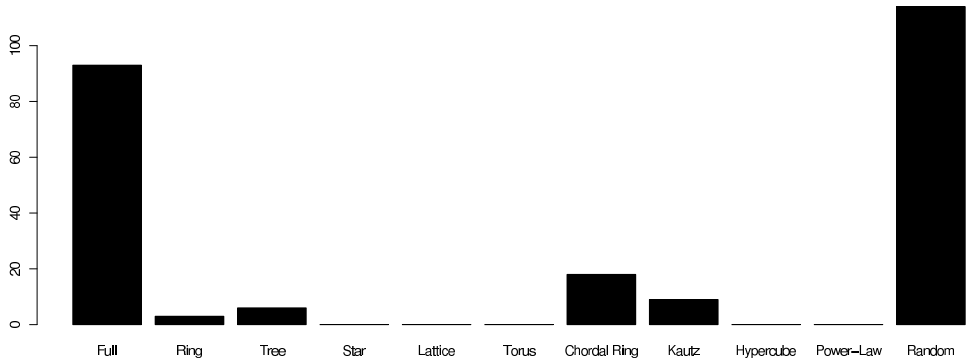


Figure 3.14: Distribution of Optimal Topology as a Function of Costs associated to Transitivity, Resilience, Average Path Length and Number of Edges.

Tree				
τ	ρ_1	ρ_2	α	ϵ
0	0	0	1	0
0	0	0	1	1
0	0	0	1	2
0	0	0	2	0
0	0	0	2	1
0	0	0	2	2

Table 3.4: Costs favoring the Tree topology.

are the only coefficients of maximization.

- The Ring is optimal when transitivity and the introduced measure of robustness are not factored into the optimization.

- The Chordal Ring appears more versatile but is often optimal when resilience is not assigned too much weight in the optimization

Ring				
τ	ρ_1	ρ_2	α	ϵ
0	1	0	2	0
0	1	0	2	1
0	1	0	2	2

Table 3.5: Costs favoring the Ring Topology

3.5 Summary

This chapter outlined a study of the structure of network topologies and presented the main differences from a distributed application perspective, between structured and unstructured topologies. The lessons learned are that the structure of the topology plays a fundamental role with respect to essential aspects of distributed applications relating to routing, search, and resilience. Experimental evaluations using simulations on a variety of topologies from regular to random, and with constant link costs of 1 were conducted. The results showed that by ranking the topologies according to *efficiency* with respect to metrics of average path length, robustness, and centrality, the fully connected network topology is the most *efficient*, succeeded by the chordal ring, the star, and the tree topologies. Future work will address arbitrary link weights distributions to better understand the optimal topology selection to match a class of applications.

Chapter 4

Network Entropy: a Measure of Neighborhood Homogeneity

4.1 Introduction

The network topology defines the "who-knows-who" relationship between network elements. These relationships can be formulated as a graph of nodes and links. The representation a network topology as a graph offers the powerful mathematical tool of graph theory that can be applied to analyze structural properties of the network. Research on network and graph theory has put forward a set of metrics that have been used to better understand properties of networks [21, 57, 10]. These metrics can be divided into either local or global depending on the proportion of the graph used to compute them. Local metrics are obtained using a neighborhood, i.e., adjacent nodes, view of a node, while global metrics are computed using knowledge of the entire graph. A summarized description of selected metrics is presented on Table 7.2 in the appendix.

Network entropy measures the expected self-information of adjacent nodes properties interpreted as random variables. It provides a measure of the homogeneity of a node, and by extension the entire graph, with respect to the *information* flowing trough the paths of the graph [58, 16]. A simple analogy to information entropy is to determine, for a given node in a graph, the number of yes/no questions that need to be asked in order to guess through which of the adjacent nodes the information is most likely to arrive from. The computation of network entropy depends on the observed property of nodes that can be either a local or global metric, for example degree when local or

shortest path related when global. The implications of using local and global network entropy measures are discussed.

This chapter first defines network entropy and goes on to present a quantitative analysis of the metric applied to graphs with static and dynamic number of nodes and edges. Network entropy is linked to local structural properties of graphs such as transitivity and assortativity.

4.2 Background and Related Work

4.2.1 Definition(s) of Entropy

There are several definitions of *Entropy* across scientific disciplines, each describing a specific property in their own field. For example, in thermodynamics, entropy characterizes the amount of energy of transformation dissipated into the environment; in statistical mechanics, it is the number of micro-configurations that can explain an observed macro-configuration; or yet, in information theory, it is the minimum number of bits required to encode a signal with known probabilities of occurrence. Clearly distinct by definition, all these instances of *entropy* point to a similar notion, a measure of *ignorance* about an observed system, which has also been commonly referred to as a measure of *disorder*.

Information entropy is a measure of the expected self-information, that is also known as *surprisal*, and corresponds to the inverse of the probability p of occurrence of a random variable, written as $\frac{1}{p}$. The more unlikely an event is to happen, the greater its surprisal. The self-information can be encoded in binary by taking the logarithm in base 2 of the surprisal, written as $log_2(\frac{1}{p})$ and is expressed in units of *bits of information*. Information entropy [59], the expected self-information, measures the average information acquired by observing a sequence of occurrences of a random variable and

is written as

$$H(x) = -\sum_{x} p_x log(p_x)$$

where p is the probability of occurrence of the observed random variable x. This definition can also be interpreted as a measure of the number of *yes/no* questions that need to be asked to find which event occurred.

The value of Entropy expressed as $H(x)$ depends on the number of observed random variables, this measure can be normalized by dividing it by the logarithm of the number k of observations $log_2 k$, and can be written as

$$H_{norm}(x) = -\frac{\sum_x p_x log_2(p_x)}{log_2 k}$$

. This formula is the one we will use to express *network entropy* in the rest of this chapter.

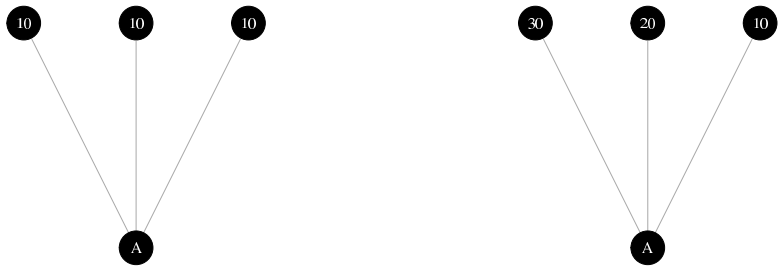
4.2.2 Network Entropy: An Illustrative Example

To illustrate network entropy, consider a graph of 4 nodes and 3 edges, such as the ones in Figure 4.1(a)4.1(b). In this example, the numbers inside the nodes are arbitrary but in practice reflect a measurable property, local or global, of the node such as for example degree, betweenness, or transitivity.

Considering a measure of entropy at node A, the network in Figure 4.1(a) has a homogeneous property distribution, while the network in Figure 4.1(b) has a heterogeneous distribution. Network entropy measures the homogeneity of a node's neighborhood with respect to a given property. The property at each adjacent node is interpreted as the occurrence of a random variable in a sequence of d draws, where d is a property of the considered node.

The normalized entropy of node A in the network in Figure 4.1(a) is 1 while the one

in Figure 4.1(b) is 0.92. This simple example illustrates how a homogeneous neighborhood is less predictable and has higher entropy, whereas a more heterogeneous neighborhood is more predictable and has lower entropy.



(a) (b)

Figure 4.1: (a) Simple Network with Homogeneous Arbitrary Node Properties. (b) Simple Network with Heterogeneous Arbitrary Node Properties.

The notion of network entropy has been studied in related work [16] using measures related to shortest paths as the observable random variables to compute the entropy metric. We briefly introduce the notion of search information, target. and road entropy proposed in related work.

4.2.3 Search Information

A stochastic search in a distributed system can be modeled by considering that a message *floods* the network starting from a source and advancing to every neighbor of that source node, which then becomes the source of the next round of propagation, with the exception that the message does not return to the node from where it originated. This model leads to a formulation of *Search Information* that is also described in [16]. Search information considers the probability of a message to reach its destination as the product of probabilities that the destination is reached through a shortest path $p(i, b)$

from a source i to the destination b, such that

$$Pp(i,b) = \frac{1}{k_i} \prod_{j \in p(i,b)} \frac{1}{k_j - 1}$$

. The factor -1 in $k_j - 1$ is due to the message not returning to its point of origin. This formulation of probability of propagation of a message along a shortest path from node i to node b leads to the information theoretic formulation of the *Search Information* as

$$S(i- > b) = -log_2(\sum_{p(i,b)} Pp(i,b))$$

.

Search information shares similarities with the self-information introduced earlier, of which entropy measures the expected value. Similarly, network entropy can be interpreted as a measure of the expected value of the search information.

4.2.4 Road Entropy

Road Entropy quantifies information when all shortest paths to and from every pair of nodes (i,j) in the network are considered and leads to an expression of network entropy as

$$R_i = -\sum_j b_{ij} log_2 b_{ij}$$

.

The values b_{ij} are related to the betweenness centrality measure [60]. Betweenness centrality provides a global measure of the importance of a node in a network by counting the number of times a node appears in the all pairs shortest paths of the network. This can be expressed as

$$C_B(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$$

, where $\sigma_{s,t}$ is the shortest path between nodes s and t.

Once computed for a static network, every node is attributed a betweenness centrality measure. As a global network measure, any modification to the network requires recomputing the betweenness.

Road entropy can then be computed for each node based on its adjacent nodes betweenness values. The more central a node, the more likely it is to carry traffic. The lower the road entropy is, the more predictable the information will be.

4.2.5 Target Entropy

Target Entropy quantifies the information at every node in the network, when considering each node as a recipient of messages signaling from all other nodes in the network. The expression for this form of entropy is

$$T_i = -\sum_j c_{ij} log_2 c_{ij}$$

. The difference with road entropy is that only those routes leading to node i are considered, the entropy is then averaged over all nodes. While road entropy measures the predictability of information *through* node i, road entropy measures the predictability *around* node i.

In [16], the authors show that when optimizing a network to minimize target and road entropy, the network is reorganized such that the predictability of information arrival is maximized. The minimal target entropy network becomes vulnerable to node attacks, whereas the minimal road entropy network becomes vulnerable to edge attacks.

4.3 Network Entropy for Varying Structural Properties

The network topology of emerging networks such as the Internet has been recognized as evolving in an ad-hoc manner. With the increasing number of deployed overlay networks offering software-level reconfigurations, it is now becoming critical to understand the properties of selected graph metrics in the context of dynamic and evolving networks in which nodes and links are added, removed, or reconfigured.

This section presents a quantitative analysis of network entropy for varying structural properties of a network topology evolved according to the Erdõs-Rényi random model. We study the impact that the operations of adding nodes, adding edges, increasing the degree correlation, and increasing the transitivity, while maintaining the network degree distribution constant [61], have on network entropy, and discuss the benefits and drawbacks of building low and high network entropy topologies. We start our analysis by quantitatively measuring network entropy on a set of known regular and random topologies.

4.3.1 Degree Network Entropy

As opposed to the Road and Target entropy presented in the previous section, we introduce the local metric of degree network entropy. The formulation of entropy remains identical to its information theoretic expression while considering the random variables as occurrences of the degrees of the adjacent neighbors. Note that in the rest of this chapter *network entropy* refers to *degree network entropy*.

4.3.2 Network Entropy of Various Topologies

While for regular topologies all nodes share a similar neighborhood structure, non-regular structures and random graphs, exhibit heterogeneous degree distribution. Figure 4.2 shows a statistical summary of network entropy for eleven considered network

topologies. Specifically, nine regular topologies: *full*, *ring*, *tree*, *star*, *2D lattice*, *2D torus*, *chordal ring*, *Kautz*, and *hypercube*; and two random: *Erdõs-Rényi* and *power-law*. The random network model has a uniform probability p for two edges to be connected of 0.2. The power-law network has a degree power-law exponent of 2.1 as has been commonly observed in real-world networks [8].

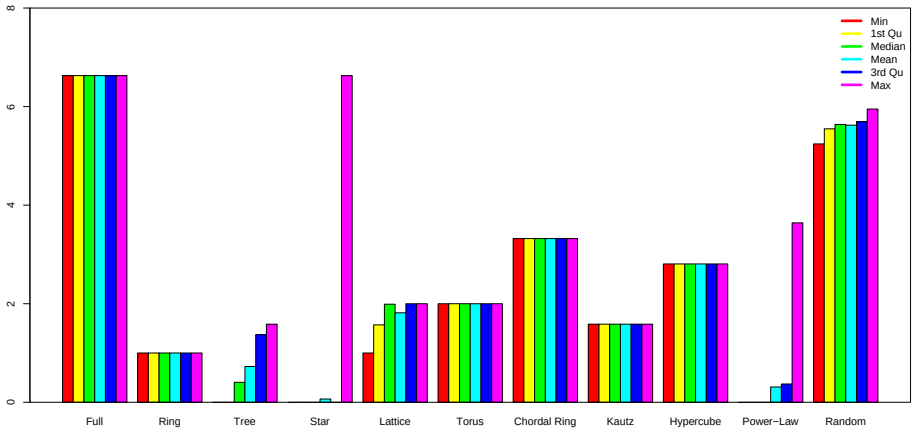


Figure 4.2: Statistical Summary of Network Entropies for Various Regular and Non-Regular Topologies

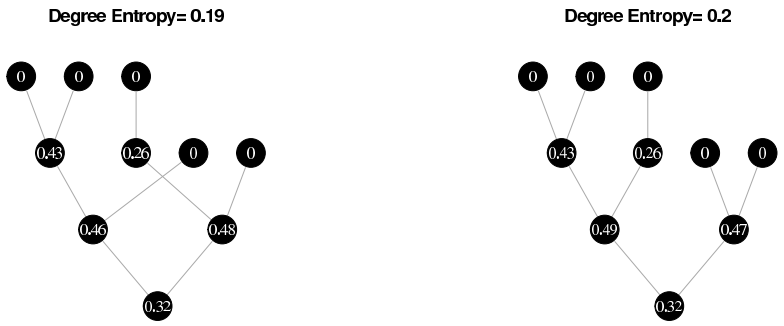
An entropy closer to 0 means more predictability and less uncertainty, therefore as we can see in Figure 4.2, the most *uncertain* topology is the Erdõs-Rényi random graph, all other topologies have negligible entropy variance.

4.3.3 Varying Structural Properties

Description of the Approach

Changing a graph structure while maintaining its degree distribution constant incurs a change in the flow of information and a change in the joint degree distribution that is reflected by the value of network entropy. For example, Figure 4.3 shows two tree networks of 10 nodes and 9 edges. The network in Figure 4.3(b) is obtained by rewiring the network in Figure 4.3(a) by swapping two edges such that the degree distribution of

the graph remains unchanged. The numbers shown inside each node correspond to the network entropy at each node, and the average for the entire network is labeled above the graph.



(a) (b)

Figure 4.3: Impact of Single Edge Rewiring on Degree Entropy.

We notice that the difference in network entropy for both graphs is different for a single edge swapping. This example illustrates the importance of structural properties for information flow, and shows that network entropy metric can be applied to quantify the predictability of information flow in the network. A low network entropy implies more certainty and neighborhood degree heterogeneity, and a high network entropy, less certainty and neighborhood degree homogeneity.

Adding Nodes

We consider a network model, such as the regular or non-regular topologies outlined earlier. We analyze the impact of varying the number of nodes on the average network entropy for a given network model. Figure 4.4 shows the results for several topologies while increasing the number of nodes between 100 to 5000 by steps of 500. The results show that while the average network entropy is different for different network topologies, it is not correlated to the variation of the number of nodes for any of the given network models. This result follows from the definition of network entropy as

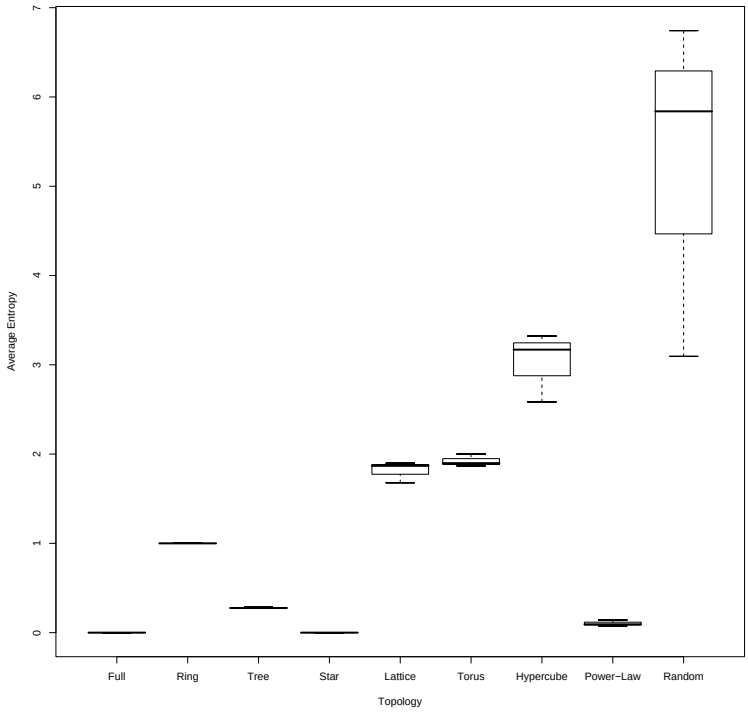


Figure 4.4: Average Entropy with increasing Number of Nodes for a Variety of Network Topologies

derived from the probability distribution of the nodes degrees, which for a given probability generating function generates a network entropy distribution function that does not depend on the number of nodes.

Adding Edges

As was shown in the previous experiment, for a given network model (i.e. identical degree distribution) the addition of nodes does not impact network entropy. However, the addition of edges at random in the network has a significant impact on network entropy as can be seen in Figure 4.5. Increasing the number of edges is associated to an increase in network entropy. This result enhances the intuitive result on network entropy of random networks, indicating that to more edges in a network is associated less predictability in information flow.

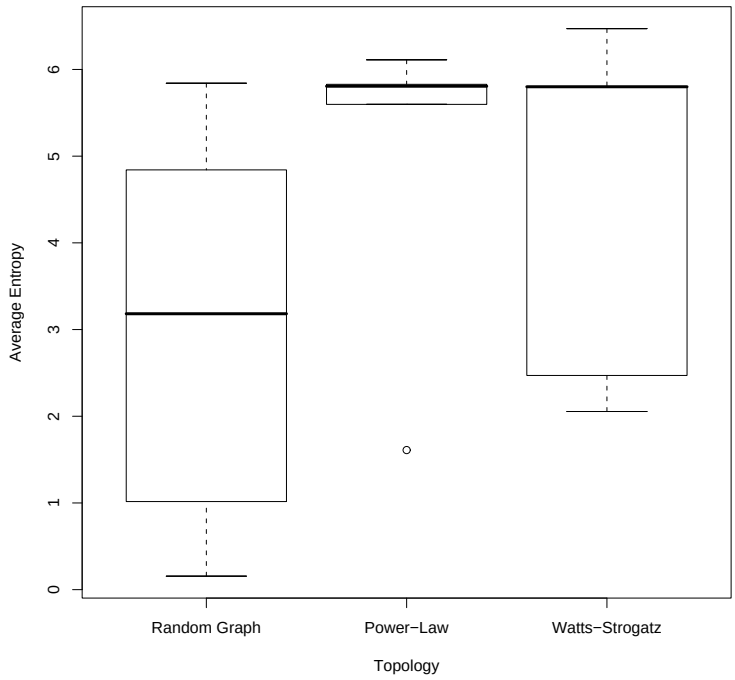


Figure 4.5: Average Entropy with increasing Number of Edges for a Random Graph Network Topology

Increasing Neighbor Degree Correlation

While the degree distribution is the most widely used structural property of a network, in the limits of large number of nodes, the possible realizations of a degree distribution is exponential and can lead to a wide variety of networks with distinct structural properties. Properties such as the average path length, diameter, or transitivity, play an important role in determining the robustness, routing performance, or security of a network. Another important property that has received attention in biological,social, and technology networks is the degree correlation between nodes of the network. A highly correlated network contains nodes with a degree that is similar to its average neighborhood degree. Uncorrelated nodes degrees appear when nodes of high degree connect to node of low degree. Finally, no correlation means that there are no apparent preferences for nodes to connect to either similar or dissimilar nodes degrees.

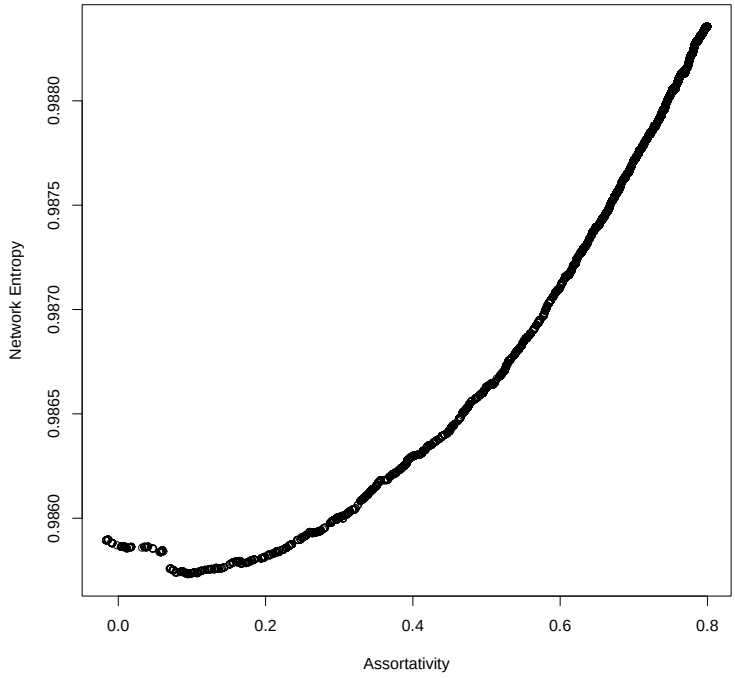


Figure 4.6: Average Entropy with increasing Network Assortativity for a Random Topology

The resulting impact of varying degree correlation on network entropy is presented in Figure 4.6. The results reveal that a highly correlated network has a higher network entropy than a low network correlation. This can be explained by the fact that high correlation is an implicit homogenization of the graph that leads to less predictability of the information flow and is reflected by higher network entropy.

Increasing Transitivity

The transitivity measures the number of neighbors of a node that are themselves neighbors. It is a characteristic network property that can be interpreted as a measure of the number of alternate paths in a neighborhood. Another interpretation of transitivity is as a measure of how far a set of nodes are from being fully connected. As can be seen in Figure 4.7, increasing transitivity of the network decreases degree network entropy. This might be caused by local increase in neighborhood degree heterogeneity formed

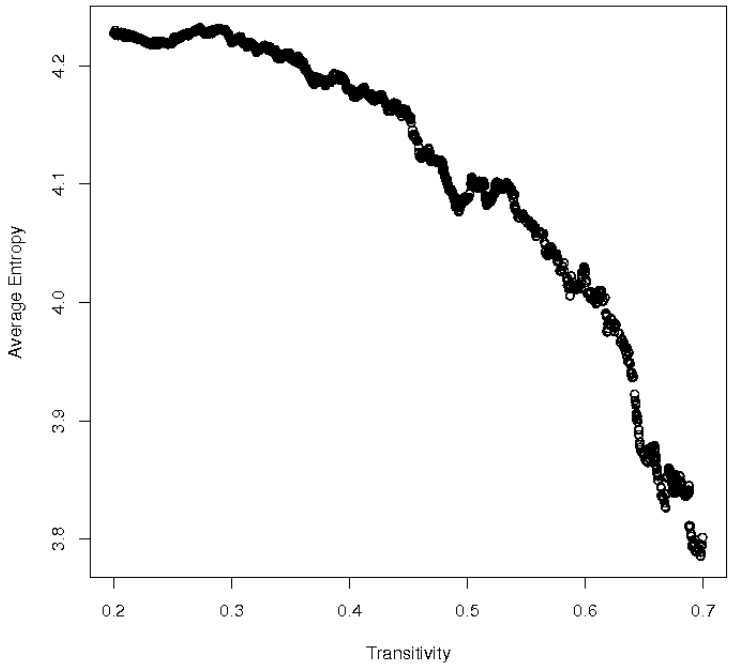


Figure 4.7: Average Entropy with increasing Network Transitivity for a Random Topology

by the increased transitivity in the network, causing the nodes to have more information to assess the source of incoming signals from its neighbors.

4.3.4 Discussion

The previous results can be summarized by observing that network entropy increases as the homogeneity of the network increases. Choosing a topology based on how *informative* the network should be can be addressed using a measure of entropy based on the degree or betweenness centrality measured at every node. Entropy can be compared to a vector of information propagation, the more informative the network is, the more spread of information potential it acquires. By analogy, if the network models an epidemic, the more *informative* networks have greater chances to successfully reveal a vector of disease than less *informative* networks.

4.4 Summary

Network entropy is a useful measure of a network property that has a strong connection with the information theoretic entropy measure. When studied as an information flow network, networks present characteristic features that can significantly affect the way in which information propagates to nodes in the network. In this chapter, we defined network entropy and studied its properties over various network topologies, from regular to non-regular. The main observations are that (1) for a given network evolution model, the average network entropy does not vary significantly under increase in the number of nodes; (2) for a given network model for which a number of edges is repeatedly added at random, the average entropy increases; (3) for a random network in which the transitivity is increased, the average entropy decreases; and finally (4) for a random network in which the assortativity is increased, the average network entropy decreases. These fundamental results show that there is a tight relationship between structural network properties and the single network entropy metric.

Chapter 5
Network Clustering

5.1 Introduction

The rapid proliferation of content produced by network applications is urging the development of new approaches to address the problem of grouping information into categories. The two main issues in addressing this problem involve defining the label of each category and the definition of algorithms to perform the actual cluster detection. The first issue relies on the derivation of standard taxonomies and meta-descriptions, while the second one addresses the underlying mechanisms and algorithms for clustering detection.

Using graph theory, a content network can be represented as a graph in which nodes represent the type of content and edges represent relationships between content types. A clustering detection algorithm operates on this graph representation to identify densely connected groups of interrelated nodes. As an example, Figure 5.1 shows a graph representation of an author network collected from the citeseer [1] online computer science bibliography database. Each node of the graph represents an author and each link represents a coauthorship on a publication. The nodes of the coauthor network are grouped into clusters that are distinctly colored.

Generally speaking, a cluster is defined arbitrarily as a set of elements that are more tightly interrelated than expected. This definition leads to the commonly observed graph theoretic definition of a cluster as a set of nodes that have more edges in common than on average. To illustrate the notion of cluster, consider for example the graph

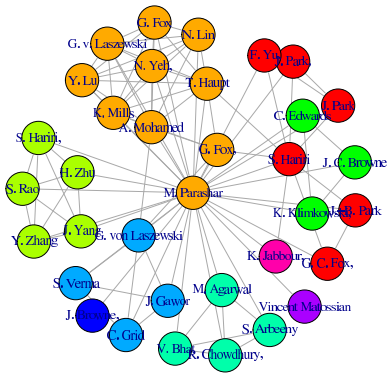


Figure 5.1: A subgraph view of the community structure of the authors based on data from Citeseer [1].

in Figure 5.2(a), the graph consists of 4 fully connected subgraphs of 8 nodes, each interconnected by one edge. Using a force-directed visual layout [2] of the graph, the 4 communities corresponding to the fully connected graphs can easily be identified by visual inspection of the graph, see Figure 5.2(b).

There are two main approaches to clustering, a top-down and a bottom-up approach:

- Top-down approaches are applied to structured environments such that the produced data is deterministically *placed* into a predetermined cluster. Examples of this approach are distributed hash tables [62] that use a set of properties of the data to map keys to value pairs.

- Bottom-up approaches are applied to unstructured environments and opportunistically detect clusters by grouping nodes by affiliation as they get discovered. Further, there are two cases in the bottom-up approach, one in which the graph is globally known a priori, and the other when the graph is only partially known through local information only.

In this chapter, we address bottom-up approaches to clustering detection and survey

a number of algorithms that address the clustering detection problem when the graph is considered static and globally known. The algorithms presented are *edge betweenness*, *greedy strategy*, *spectral partitioning*, and *random walker*.

Applications of identifying clusters in unstructured P2P overlay network applications are addressed. In particular, the application of cluster detection as a mechanism to create a *compressed representation* of a graph that can be exchanged amongst nodes in the network to improve information discovery is presented. Finally, we propose, evaluate, and compare an algorithm to detect clusters that is based on partial information. The approach uses the network entropy 4 of nodes and is conceptually similar to the edge betweenness clustering detection algorithm. Network entropy cluster detection differs from all other approaches by relying exclusively on local information. The algorithm runs in $O(Nk)$ where N is the number of nodes in the graph and k is the maximum node degree. The results show that the local approach based on network entropy is fast and performs well compared to the global solution on two modeled, random and power-law degree distributed, and two real-world networks, the Canadian Autonomous System and the Gnutella networks.

5.2 Goodness of Clustering and Overlay Networks[1]

The clustering detection problem consists of identifying the minimum number of edges that divide the network into distinct clusters. Real networks do not resemble the modular networks such as the one shown in Figure 5.2(a). Therefore, the goodness of the partition of a network into clusters is an arbitrary measure that is often hard to quantify. This quantification has been addressed using the modularity metric [63] as a statistical

[1]Many approaches to clustering described in this section are derived from the literature in statistical physics that refers to this type of clustering as community. We will refer to clustering as community when necessary to maintain the terminology consistent with the way it is published in the literature.

measure of the quality of a graph division. Modularity has been widely accepted by researchers as a measure of the goodness of a network division into clusters, a higher modularity reflecting a better partitioning.

5.2.1 Modularity

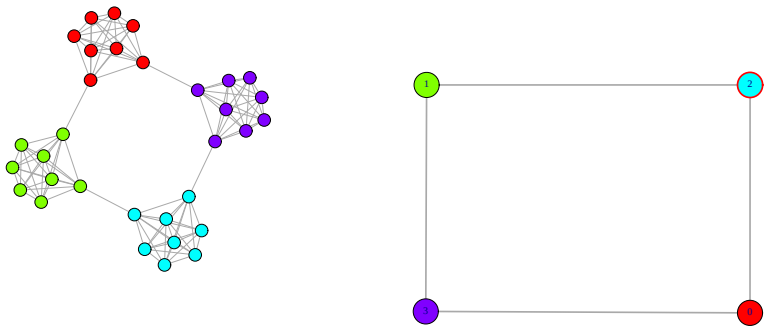
Modularity [63] measures the quality of a graph division into communities. Modularity is a statistical measure of the expectation for an edge to be inside a community rather than between communities. The idea underlying modularity is that if an edge inside a community is removed it contributes little to the modularity whereas when an inter-community edge is removed it contributes much. Let e_{ij} be the fraction of edges in the network that connect vertices in group i to those in group j, and let $a_i = \sum_j e_{ij}$ be the degree of vertex i. Then the modularity Q is expressed as

$$Q = \sum_i (e_{ii} - a_i^2)$$

. It is a measure of the fraction of edges that fall within communities minus the expected value of the same quantity if edges fall at random without regard to the community structure. Modularity is computed using a mapping of nodes to community memberships. A membership assignment that results in higher modularity reflects a better division of the network into communities.

5.2.2 Clustering and Peer-to-Peer Overlay Networks

In a physical topology, the edges represent the connectivity between nodes physically connected by a link, in contrast, in an overlay network topology the relationships between nodes can reflect arbitrary properties of the nodes and links such as content, location, affinities etc... For example in Figure 5.2(a), the fully connected subgraphs might represent 4 physically interconnected local area networks in a wide area network, while as an overlay topology it might represent the grouping of data generated by a sensor network that has been tagged as 4 distinct, yet interrelated, categories.



(a) (b)

Figure 5.2: (a) A Modular Graph of 4, 8-node fully connected subgraphs. (b) Communities of the graph of the network shown in (a).

The virtualization of overlay networks makes clustering detection a useful application to aggregate relationships into clusters in a dynamically and adaptive manner [62]. Using cluster detection mechanisms, the overlay network becomes an ad-hoc searchable relational database.

5.3 Survey of Algorithms for Cluster Detection

This section addresses cluster detection for static and globally known graphs. We present a brief survey of algorithms that have been proposed to detect clusters in such graphs. Most work in this area has been done as part of inter-disciplinary studies in statistical physics, social science, and computer science. The algorithms that we present are based on distinct strategies that work more or less well depending on the size and sparsity of the graphs considered.

The algorithms presented are based on *edge betweenness*, *greedy strategies*, *spectral partitioning*, and *random walker*. All these algorithms aim to achieve the same objective, that of identifying in as few steps and minimal computational cost as possible the edges separating the graph clusters[1].

5.3.1 Edge Betweenness Community Detection

The betweenness centrality measure was proposed by Mark Newman [5] as the number of shortest paths that go through a vertex (vertex betweenness) or edge (edge betweenness) of a graph. It is intuitive to understand that the most traveled vertex or edge is more central than a rarely traversed vertex or edge.

The approach to detect communities based on this metric consists in consecutively removing the edge with highest betweenness centrality. The process is then repeated for each new graph that results from the edge removal, and is performed as many times as the number of desired communities to detect is reached or until all edges have been removed. This is a very reliable approach that yields accurate community detection, but has the down side of being very costly. Every betweenness computation requires an all pairs shortest paths computation that requires $O(n^3)$ operations, this can be done in worst case $O(n$ times, yielding an $O(n^4)$ computation cost, therefore making the approach intractable for very large graphs.

5.3.2 Greedy Strategy

The greedy strategy relies on the concept of *Modularity* proposed in [63]. This approached is based on the idea that starting from a graph with N nodes and N communities, each node attempts to form a community with any of its m neighbor nodes, a successful merge between two nodes is the one that maximizes the value of the modularity. Therefore at each step of the algorithm the cost to compute the modularity is $O(n)$ which is performed $O(n)$ times, leading a total cost for the greedy approach of $O(n^2)$. The greedy strategy is fast compared to other approaches and works well in most cases but is not as reliable and accurate as the edge betweenness community detection.

5.3.3 Spectral Partitioning

A matrix representation of a graph is one in which the relationships between vertices are expressed in matrix form as entries m_{ij} between vertices i and j representing a particular characteristic of their relationship. For example, in an adjacency matrix representation, a_{ij} is 1 if an edge between the vertices exists and 0 otherwise. In a Laplacian representation, entry l_{ii} has the degree of node i, and entries l_{ij} where $i \neq j$ is -1 if there is an edge between i and j and 0 otherwise. The modularity introduced in the previous section, can be represented as a matrix where q_{ij} is $a_{ij} - p_{ih}$ where p_{ij} is the probability that there is an edge between vertices i and j, and a_{ij} corresponds to the entry of the adjacency matrix.

Using the matrix representation of a graph, spectral approaches are concerned with the decomposition of the matrix into eigenvalue and eigenvector form. A common practice in graph partitioning is to minimize the *cut size* between two groups of vertices. The cut size can be expressed as

$$R = \frac{1}{2} \sum_{i,jindifferentgroups} A_{ij}$$

and further reduces to

$$R = \frac{1}{4} s^T L s$$

where L is the Laplacian matrix, that can finally be written as

$$R = \sum_i a_i^2 \lambda_i$$

where λ_i is the eigenvalue of vertex i and $a_i = v_i^T s$ is an expression of the eigenvector v_i of L. Therefore minimizing R involves finding the values of a_i that place as much weight as possible on the smallest eigenvalues λ_i [64].

Similarly, using the modularity matrix representation

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - P_{ij}] \delta(g_i, g_j)$$

where $\delta(r, s) = 1$ if $r = s$ and 0 otherwise, and m is the number of edges in the graph. This can be reduced to the form

$$Q = \frac{1}{4m} s^T B s$$

, where $B_{ij} = A_{ij} - P_{ih}$ is the modularity matrix. Using the eigenvalue representation, B can be expressed as

$$B = \frac{1}{4m} \sum_i a_i^2 \beta_i$$

and would need to be optimized rather than minimized as in the previous case of the cut size.

Once partitioned into two distinct subgroups, the process is repeated until no further contributions to the modularity is noticeable. The cost of running this algorithm lies on the cost of identifying the eigenvectors of the matrix representation, which has a worst case cost of $O(n^3)$ when using a non-optimized Lanczos method on a given matrix, but can be reduced to $O(n^2)$ for sparse graphs using optimized algorithms. Therefore the total cost of the spectral approach is on the order of $O(n^2)$.

5.3.4 Random Walker

Random walkers are conceptual agents that traverse a graph by stochastically selecting their next hop. The intuition underlying the approach based on *random walkers* is that by walking over the graph's paths, a walker given enough time will eventually spend more time within a community than outside of it. This process is explained in detail and formalized in [65]. A set of random walkers are initialized on a graph and decide which path to take at random for a given number of steps, the length of the random walk. After all walkers have completed their round, the algorithm counts for each pair of nodes, i.e. edge, the number of walkers that traversed the given edge. If the number of walkers that traversed the edge is much greater than the number of walkers that traversed the endpoints on their way to other destinations, then the edge is more likely to be an inside community edge rather than an in-between community edge, therefore the

two nodes are merged into a single community.

The random walker is therefore a statistically based variation on the edge betweenness community detection algorithm for which every possible shortest paths is calculated. The cost of the random walker algorithm depends on the length of the walk but is in the worst case of the order of $O(mn^2)$ where m is the length of the walk.

5.4 An Approach to Cluster Detection based on Network Entropy

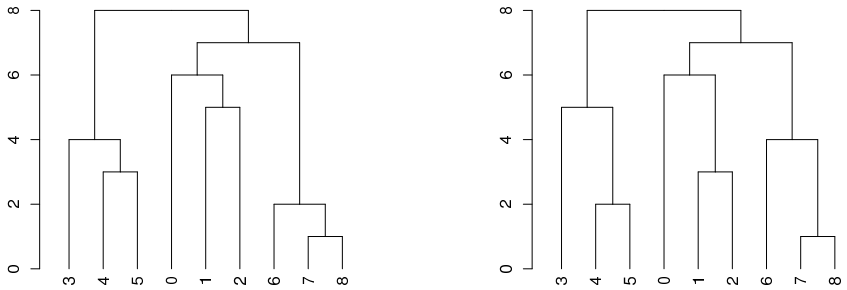
We propose an approach to cluster detection that uses local information aggregated as *network entropy* to assess a global measure of *importance* of a link. The intuition underlying the network entropy approach is that the two endpoints of an often traveled link, i.e., high betweenness link, might have more uncertainty regarding which node information might be coming from than the nodes that connect to the endpoints. These neighboring nodes might indeed have low entropy as they are more *certain* that information would be traversing the high entropy endpoint.

Algorithm 2 Cluster Detection Using Network Entropy

Require: Graph: G(E,V); Edges: E; Nodes: (i,j) $\in V$
  while true **do**
    for all $E(i,j) \in G$ **do**
      $N_{G,i} \longleftarrow = DegreeEntropy(G,i)$
      $N_{G,j} \longleftarrow = DegreeEntropy(G,j)$
    end for
    for all $E(i,j) \in G$ **do**
      $N_{i,j} = N_{G,i} + N_{G,j}$
    end for
    edge.to.remove=which.max($N_{i,j}$)
    edges $\leftarrow$ append(edges,edge.to.remove)
    G =delete.edge(G,edge.to.remove)
    if number.of.edges(G)==0 **then**
      return edges
    end if
  end while

The algorithm for cluster detection using network entropy 5.4 computes the degree

entropy at every node, an $O(Nk)$ operation, where N is the number of nodes and k the degrees of the nodes. Then for each edge, the network entropy of the two endpoints are added and sorted in decreasing order, such that the highest entropy reflects the edge for which endpoints reflect the most uncertainty. This edge is then considered to be a dividing edge and is removed from the network. In a process similar to the edge betweenness approach, this operation is repeated until all edges have been removed. The clusters are then merged starting from the last removed edge all the way to the first dividing edge.



(a) (b)

Figure 5.3: (a) Edge Betweenness Community Formation Tree. (b) Network Entropy Cluster Formation Tree.

A result for the network in Figure 5.2(a) is presented in Figures 5.3(a) and 5.3(b) as a dendrogram plot. This plot shows the successive network merges that the algorithm identified and for which each merge is a step in the progress of the algorithm. The Figures show that the clusters identified by the edge betweenness based algorithm and network entropy based algorithm are exactly similar. Results on various degree distributed graphs will be presented in Section 5.6.

5.5 Evaluation

We consider a set of four network topologies, two network models, a random graph evolved using the Erdõs-Rényi process, and a power-law degree distributed network evolved using the Barabási-Albert process, and two real-world networks, one of the Canadian Autonomous System and the other from the Gnutella Peer-to-Peer network. We evaluate the community detection algorithms on each network and report the results in Figure 5.4 in which the networks are respectively labeled ERG, BAG, Canada, and Gnutella.

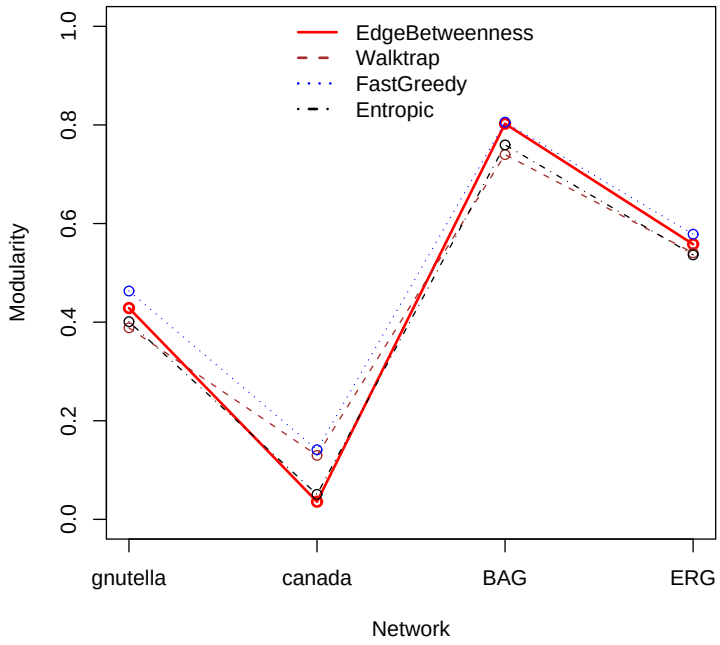


Figure 5.4: Comparison of various approaches to cluster detection.

The performance of the partitioning is measured using the modularity metric introduced in [63]. All four cluster detection algorithms, *edge betweenness*, *walktrap*, *fast greedy*, and *entropic*, show comparable results. The surprising result is that the entropic community is the only one to consider local information alone, and shows comparable results to global strategies. Figure 5.5 shows a dendrogram plot of the

clustering process applied to power-law degree distributed scale-free network with 125 nodes, and illustrates the differences between each clustering algorithm. The figures show how nodes are identified as belonging to a cluster, for which a horizontal cut in the dendrogram corresponds to the number of clusters at a given step of the algorithm.

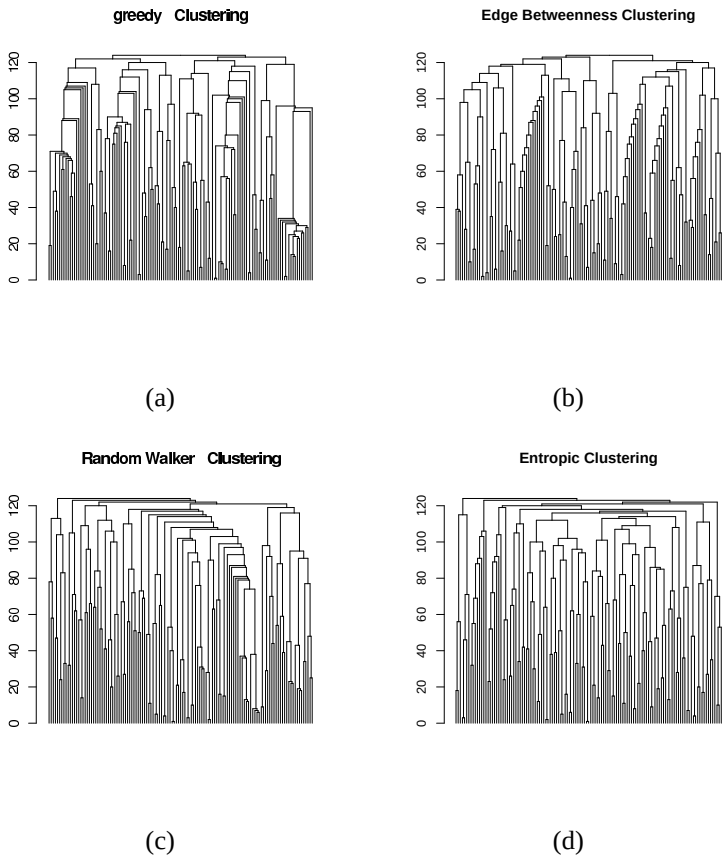


Figure 5.5: Dendrogram representations of various clustering detection algorithms. (a) Greedy Clustering. (b) Edge Betweenness Clustering. (c) Walktrap clustering. (d) Network Entropy Clustering.

5.6 Applications

Cluster detection is a powerful mechanism to identify partitions in a large network graph. The applications of such a mechanism are many and encompass techniques of graph partitioning for distributing load and computation on parallel and distributed compute resources, compressing a large graph into a reduced form, i.e. snapshot, that

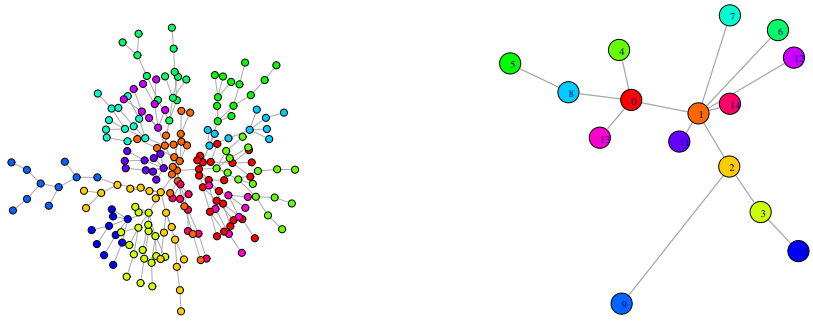
can be easily shared amongst nodes in the network, assist in visualization of large data sets by providing a hierarchical field of view from macro to micro modes, or even identifying clusters of interest in social network applications.

5.6.1 Graph Partitioning for Parallel and Distributed Computing

A partition of a graph is one that divides the nodes into distinct sets or groups. The applications of graph partitioning to parallel and distributed computing has been concerned with the identification of the minimal cuts that separate the graph into equally distributed sets, be it computational resources or computational loads. The identification of clusters is not so much concerned with partitioning the graph into a balanced partition of the graph, rather than identify through clustering the areas that share most affinities. This approach can therefore show that a large portion of a computation is spent on a very small fraction of the resources and can assist in devising a solution to balance the workload across the available resources. Identified clusters therefore act as a snapshot of the entire graph and are necessary to support further strategies or approaches to deal with the application's objective.

5.6.2 Graph Compression

Detecting clusters is a way to compress large graphs by merging densely connected subgraphs under the label of a single cluster. For a sparse graph with many such densely connected subgraphs, the overall graph can be significantly reduced into a graph of clusters and be of great help to assist in visualizing very large graphs, such as in the order of thousands of nodes. To illustrate this point, consider for example a small to medium 200 node power-law degree distributed network as presented in Figure 5.6(a). Using a greedy strategy, the 15 clusters are detected and are shown in Figure 5.6(b). While the original graph consists of 200 nodes and 199 edges, the compressed graph comprises 15 nodes and 14 edges, a compression ratio of 92.5%!



(a) (b)

Figure 5.6: (a) 200 Node Power-Law Network. (b) 15 Node compressed power-law network

As the compression ratio is dependent on the number of clusters detected, we are interested in arranging a topology of given degree distribution such that the number of clusters is either minimized or maximized.

5.6.3 Clustering of Various Network Topology Models

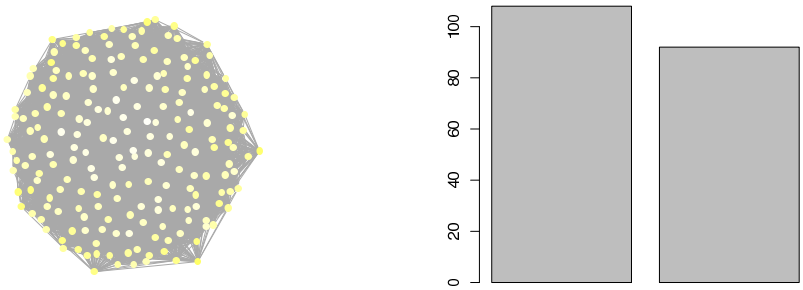
Following the observation that clustering can be used as a compressed representation of a large graphs, it is relevant to answer the question: *what is the distribution of cluster sizes of various topologies?*

To this end, we present three cluster distributions from real-world networks, *Uniform*, *Poisson*, and *Zeta*. Note that while it is obvious that the ultimate compression rate of a graph is 100% corresponding to reducing a graph to a single node, this is not of any practical value.

Uniform distribution: A probability distribution is uniformly distributed when all events out of k events have exactly the same probability of occurrence $\frac{1}{k}$. Applied to a degree distribution, the uniform distribution implies that all degrees are equal and therefore yields a regular structure such as *rings*, *tori*, or *hypercubes*. Given the regular pattern of such a distribution, the identification and labeling of a clustering

depends on the granularity at which the clustering is desired. The compressed view of a uniformly distributed network is an identical structure at a lower dimension and therefore corresponds to a constant and preset compression rate.

Poisson distribution: describes the number of unlikely events that happen within a certain time. This distribution is characterized by an exponential distribution centered around an average number of occurrences λ and is expressed as $f(k, \lambda) = \frac{e^{\lambda}\lambda^{k}}{k!}$, where k is the number of events observed. The Poisson distribution naturally emerges in a network where the probability p of connection between any two vertices is the same for every pair of nodes (although this is more accurately represented by a Bernoulli distribution, which for large number of occurrences leads to a Poisson distribution). This strategy leads to a total number of edges $\frac{n(n-1)p}{2}$ where the dividing factor of 2 is for the case of an undirected graph in which every edge is counted twice. Figures 5.7(a) 5.7(b) shows respectively a sample Poisson distributed random graph of 200 nodes and the associated distribution of clusters detected. The compression rate in this case is 99%.

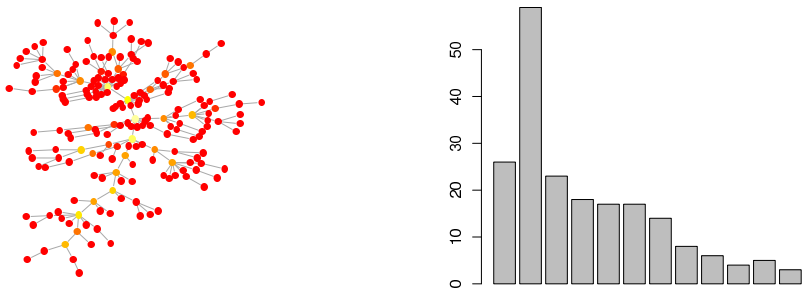


(a) (b)

Figure 5.7: (a) A sample Poisson degree distributed network. (b) Corresponding clusters detected using network entropy based clustering detection algorithm

Zeta distribution: obeys a power-law in which the rank of an observed occurrence k is $\frac{1}{k^{\alpha}}$, with α the exponent of the distribution. This distribution has been observed

across a variety of systems such as the distribution of wealth in a country, actor networks, and web graphs. The *scale-free* network that evolves through a model of preferential attachment as proposed by Barabasi and Albert exhibits such power-law degree distribution and is widely recognized model of evolving network with such characteristic. Figures 5.8(a) 5.8(b) show respectively a sample power-law distributed graph of 200 nodes and the associated distribution of clusters detected. The compression rate in this case is 93.5%.



(a) (b)

Figure 5.8: (a) A sample power-law degree distributed network. (b) Corresponding clusters detected using network entropy based clustering detection algorithm.

5.7 Summary

Cluster detection on dynamic graphs is becoming increasingly important as data generating networks seek to find clustering solutions using distributed algorithms rather than the aggregation of global information at a single node. Such approaches are critical in sensor network applications in which the sensor data should be aggregated at a data layer to provide the scientists and engineers with a relational data querying mechanism to assist in the data mining process.

This chapter motivated the problem of cluster detection in large networks.A novel approach based on a local metric of network entropy is presented and shown to perform comparatively well to global methods on a variety of graphs. Applications of cluster detection on static and globally known graphs is an extremely valuable tool that can be used for visualizing large datasets as compressed graphs. Further, the identification of clusters and the distribution of cluster size provides a more robust and practical result than the more common graph partitioning techniques that are concerned with identifying the bisecting edge or bisection bandwidth.

Chapter 6

Evolving Topologies with Arbitrary Structural Properties

6.1 Introduction

An increasing number of unresolved problems in the sciences are being formulated as complex networks [5, 17]. From protein folding in biology to population migrations in social science, modeling dynamic adaptive processes can often be reduced to the identification of structural properties, i.e., topology, of a graph representation of the actors at play. Large-scale computer networks, such as the Internet have recently been recognized as complex networks. The network topology of such emerging networks is the result of locally adaptive processes that have an impact on application-level functionalities. These local processes may be triggered by administrative policy or engineering requirements and consist of nodes and links being added, removed, or reconfigured dynamically. At the application-level, the topology plays a determining role with respect to routing, search, robustness, and clustering.

Understanding the effect of these local adaptations on the global structure of the network is an increasingly important problem, partly due to the exponential growth of networks and partly to the software-level configuration offered by overlay network topologies. For example, consider the growth of the Internet's Autonomous Systems. Each Autonomous System can define unique routing or behavioral policies (i.e., content publication, censorship) that are propagated to other Autonomous Systems. The recent increase in the addressing number of the Autonomous System from 16 bits to 32

bits suggests that more policies are likely to be deployed in the near future and therefore call for an urgent need to first, understand the structure of the networks formed, and second to control the dynamics of such potential rapid growth. In the absence of control, such networks could be subject to increased vulnerability as well as to planned or unplanned attacks possibly leading to a level of disruption that the modern computerized world's economy could significantly suffer from [12]. As another example, Peer-to-Peer data sharing applications are overlay network topologies that have gained wide popular appeal in the past decade, and established themselves as efficient modalities for large volume data transfer [66]. The rapid and constant growth in publication of content over the Internet requires overlay networks to form application-level connections that speed search and transfer time.

This chapter considers six network reconfiguration strategies that are based on local metrics of the graph representation of the network. The strategies consist of reconfiguring the network towards maximizing and minimizing *assortativity*, *transitivity*, and *entropy*. An analysis of the impact of these optimizing strategies with respect to application-level properties of *average path length*, *search*, *robustness*, and *clustering* is presented.

Experimental evaluations of the application of these optimizing reconfiguration strategies are conducted on two topology models, one with Poisson and the other with a power-law degree distribution, and two real-world networks, Gnutella Peer-to-Peer network and the Canadian Autonomous System.

6.2 Network Graph Models

We consider two network models, a G_{np} Erdõs-Rényi model and a scale-free network model of Barabási-Albert.

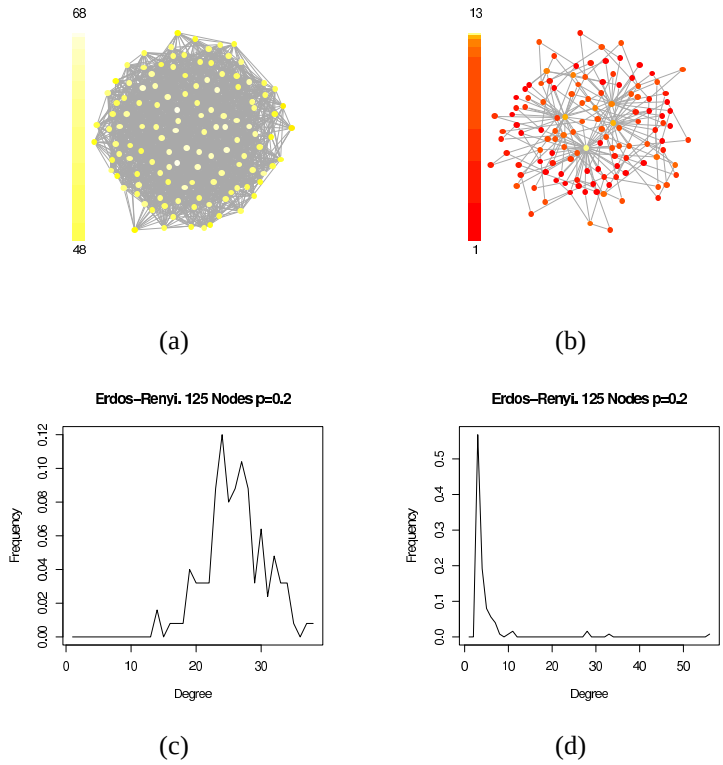


Figure 6.1: (a) Erdõs-Rényi random graph model. $\{p = 0.2$; 125 Nodes; 1564 Edges$\}$. (b) Barabási-Albert scale-free network $\{\alpha = 1$; 125 Nodes; 248 Edges$\}$. The color bar besides each network corresponds to the range of the degree entropy for every node in the network. (c) Degree Distribution corresponding to random graph. (d) Degree Distribution corresponding to scale-free network.

The network models are used to evolve two network topologies, a random network shown in 6.1(a) and a scale-free network shown in Figure 6.1(b). The generating principle that drives the evolution of the random network yields the expected Poisson degree distribution as shown in Figure 6.1(c). The Barabási-Albert network evolved in discrete time steps using a preference function of $\alpha = 1$ that translates into a power-law exponent in the degree distribution of the resulting network, as shown in Figure 6.1(d).

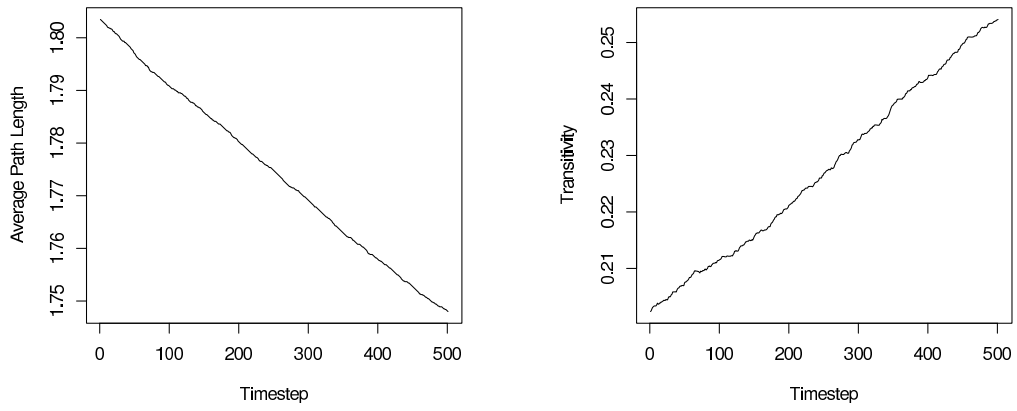
6.3 Random Link Addition

The small-world network model proposed by Watts and Strogatz [41] formalizes the idea that adding random links in a network yields networks with shorter average path

length and increased transitivity. We experimentally iterate on these theoretical results for the two considered topologies.

6.3.1 Random Network

The results of the random link addition for the $G(125, 0.2)$ Erdõs-Rényi random network are shown in Figures 6.2(a) and 6.2(b).



(a) (b)

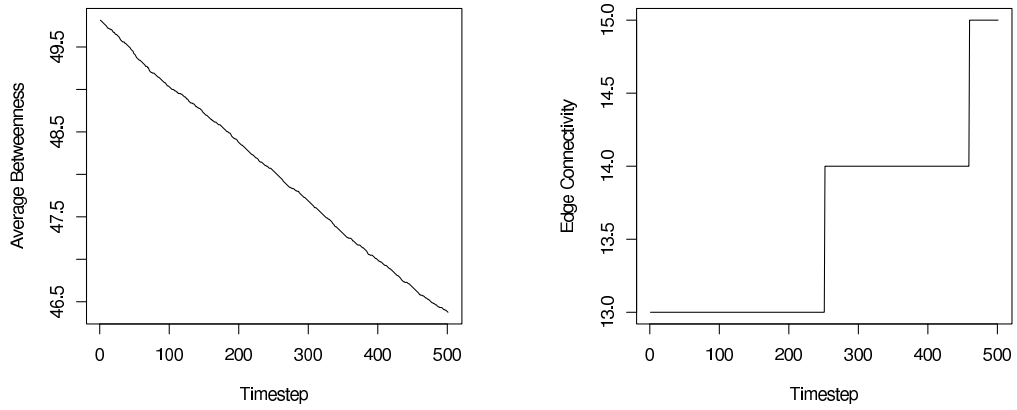
Figure 6.2: (a) Effect of Random Link Addition on Average Path Length and (b) on Transitivity for $G_{125,0.2}$.

Further, we show that adding links increases the robustness of the network by increasing the edge connectivity as well as decreasing the average betweenness for the considered network models, as can be seen in Figures 6.3(a) and 6.3(b).

6.3.2 Power-Law Network

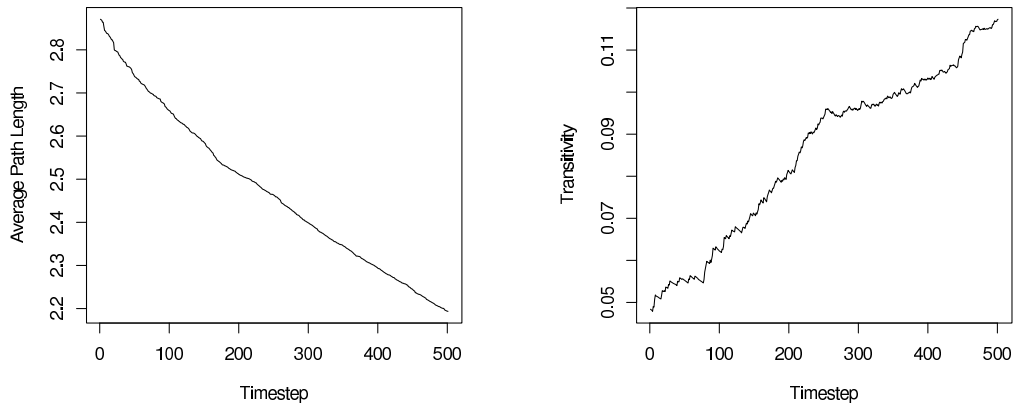
The results of the random link addition for the Barabási-Albert preferential attachment networks are shown in Figures 6.4(a) and 6.4(b).

As can be seen in Figures 6.5(a) and 6.5(b), similar to the Random network model, adding links increases the robustness of the network by increasing the edge connectivity as well as decreasing the average betweenness.



(a) (b)

Figure 6.3: (a) Effect of Random Link Addition on Average Betweenness Centrality and (b) Edge Connectivity for $G_{125,0.2}$.



(a) (b)

Figure 6.4: (a) Effect of Randomly Adding Links on Average Path Length and (b) on Transitivity for scale-free network.

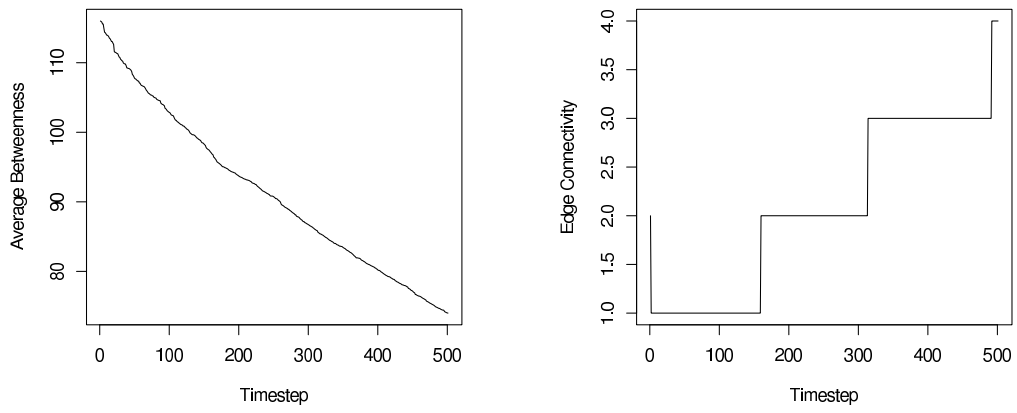


Figure 6.5: (a) Effect of Random Link Addition on Average Betweenness Centrality and (b) Edge Connectivity for scale-free network.

6.3.3 Local Metrics

The graph theory literature has formulated metrics aimed at assessing a wide range of properties of graphs. These metrics can be directly applied to networks represented as graphs, and can be broadly categorized into either local or global depending on the proportion of the network required to be computed. The only strictly structural property of a node in a graph is its degree, which corresponds to the number of neighbors that a node has. We identify canonical local metrics derived from node degree that take into consideration the structure of a node's neighborhood. In particular, we consider *transitivity*, *assortativity*, and *entropy*.

Transitivity: measures the number of neighbors of a node that are themselves neighbors. It can be measured as the fraction of triangles in the neighborhood of a node with respect to the total possible number of triangles that could be formed[1]. In other words, transitivity measures how far from fully connected a node's neighborhood is.

[1]This formulation of transitivity is the one used in social network analysis. Another measure of transitivity is described and compared in [5].

Assortativity: measures the homogeneity between a node's degree and its neighborhood's degree. Positively correlated nodes have similar degrees and are said to be assortative, while anti-correlated nodes degrees have dissimilar degrees and are said to be disassortative.

Degree Entropy: measures the homogeneity of the degree of a node's neighborhood. Similarly to transitivity but unlike assortativity, entropy does not consider the degree of the node at which it is computed as part of its formulation. A low entropy is associated to a heterogeneous neighborhood degree. The more homogeneous a node's neighbors degree, the higher the entropy.

These three metrics are considered because they reflect properties of the network that respectively relate to robustness, homogeneity, and information.

6.3.4 Evolving Networks with Arbitrary Structural Properties

The effect of randomized link additions presented in the previous section showed a significant impact on global properties of the network models considered. We now turn our interest to modifications of the network that maintain the degree distribution constant, i.e., each node has a fixed number of neighbors, and study the impact of these modifications at the application-level.

The properties of assortativity, transitivity, and entropy for the networks considered are shown in Table 6.1.

Transitivity, Assortativity, and Entropy reflect canonical local properties of a network topology that, along with degree distribution, provide a *signature* of a network. Finding a lower and upper bound on these metrics for an arbitrary network while maintaining the network connected is an NP-complete problem. In order to empirically

Graph	Transitivity	Entropy	Assortativity
Erdõs-Rényi	0.2023626	0.6613184	-0.01692184
Barabási-Albert	0.0483871	0.1580675	0.1728174

Table 6.1: Properties of the two considered graphs, Erdõs-Rényi random graph and Barabási-Albert scale-free network.

determine these bounds we use a simulated annealing optimization on the considered networks.

Simulated Annealing

The networks are modified using a simulated annealing and Metropolis-Hastings algorithm. This method, shown in Figure 6.6, consists of selecting two edges at random, such as (A, B) and (C, D), in the network and exchanging them resulting in edges (A, D) and (B, C) such that the degree distribution of the graph is maintained. The selected edges are only considered if they are strictly independent, i.e., the potential target node is not in the source node's neighborhood. The structural properties of interest are then measured on the resulting candidate network and is accepted if, (1) the network remains connected and (2) it has an improved (according to the desired property) measure. We use an annealing optimization process such that a candidate non-optimizing move is also accepted with probability $e^{-\beta x}$ in which x is the annealing schedule and β a multiplicative tolerance coefficient.

$$P(accept) = \begin{cases} 1 & \text{if move is optimizing} \\ e^{-\beta x} & \text{if move is not optimizing} \end{cases}$$

6.4 Univariate Network Optimizations

In the following set of experiments, each property is optimized independently using a univariate simulated annealing optimization. The results are presented in Tables 6.2 and 6.3. The results are reported in tables that list the title of the graph in the first row

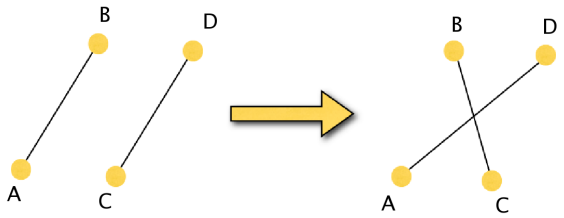


Figure 6.6: Rewiring operation maintains degree distribution constant.

and first column, the name of the metric measured on the network is reported in the column title, the name of the strategy used to optimize the network is reported in the first column of every row. The name of the networks for the random and scale-free networks are respectively reported as ERG and BAG. The strategies that maximize the local metrics are those that maximize Transitivity, Assortativity, and Entropy, labeled respectively *MaxTr, MaxAs, MaxNtrp,* and those that minimize the local metrics are respectively labeled *minTr, minAs, minNtrp*.

6.4.1 $G_{(125,0.2)}$ Erdõs-Rényi Random Graph

Table 6.2 presents the results of the optimizations for the considered Erdõs-Rényi random graph. These results set optimal bounds on the networks resulting from the optimizations. The assortativity measure seems to be efficiently optimized, in the range of $[-82.94\%, +81.54\%]$, while other metrics reflect more modest optimizations. Notably, the degree network entropy has a very small window of optimization, from $[-0.04\%, 0.24\%]$, that is due to the fact that degree network entropy is increased only when a node connects to a homogeneous neighborhood, which is already the case for the initial $G_{(n,p)}$ random graph, and leaves lesser opportunity for optimization.

Graph	Transitivity	Entropy	Assortativity
ERG	0.2 [0 %]	0.66 [0 %]	-0.02 [0 %]
MaxTr	**0.53 [32.5 %]**	0.66 [0.03 %]	0.13 [14.96 %]
MaxAs	0.43 [22.91 %]	0.66 [0.22 %]	**0.8 [81.54 %]**
MaxNtrp	0.08 [-11.81 %]	**0.66 [0.24 %]**	-0.85 [-83.38 %]
minTr	**0.04 [-16.07 %]**	0.66 [0.01 %]	-0.13 [-11.75 %]
minAs	0.09 [-11.67 %]	0.66 [0.23 %]	**-0.85 [-82.94 %]**
minNtrp	0.29 [8.98 %]	**0.66 [-0.04 %]**	0.02 [3.98 %]

Table 6.2: Univariate Optimization of Assortativity, Transitivity, and Entropy of a Random graph.

6.4.2 Barabási-Albert Graph

The results for the power-law degree distribution of the scale-free networks of Barabási and Albert shown in Table 6.3 reveals optimizations of the local metrics that are significantly different than the Poisson degree distribution of the previously considered random graph. Transitivity is unchanged because the model generates an acyclic network. Degree network entropy is optimized between $[-1.57\%, 2.67\%]$ and assortativity between $[-54.47\%, 52.67\%]$. Compared to the previously described random graph, the fact that few nodes of the scale-free network have high degree and many low degree widens the window of optimizations, the maximum assortativity is therefore decreased, but the degree network entropy optimizations wider, due to more degree *diversity* in the network.

Graph	Transitivity	Entropy	Assortativity
BAG	0 [0 %]	0.03 [0 %]	0.03 [0 %]
MaxTr	0 [0 %]	0.04 [0.31 %]	0 [-2.88 %]
MaxAs	0 [0 %]	0.03 [-0.8 %]	**0.56 [52.67 %]**
MaxNtrp	0 [0 %]	**0.06 [2.67 %]**	-0.37 [-40.31 %]
minTr	0 [0 %]	0.04 [0.22 %]	-0.01 [-3.99 %]
minAs	0 [0 %]	0.06 [2.55 %]	**-0.51 [-54.47 %]**
minNtrp	0 [0 %]	**0.02 [-1.57 %]**	0.11 [8.33 %]

Table 6.3: Univariate Optimization of Assortativity, Transitivity, and Entropy of a scale-free graph.

6.5 Effect of Local Perturbations on Application-Level Properties

Unlike local node properties, global properties of a network require computation on the entire network to quantify the property at each node. The networks obtained by optimizing on local properties have distinct local and global characteristics from the original networks. The global metrics of choice in our study relate to fundamental characteristics of any network application, measures of path lengths, robustness, and clustering.

The application-level metrics considered are:

- **Average Path Length**: measures how long a path is on average between all pairs of nodes in the network.

- **Search Information**: considers the probability of a message reaching its destination as the product of probabilities that the destination is reached through all degenerate shortest paths $p(i,b)$ from a source i to the destination b, such that

$$P\{p(i,b)\} = \frac{1}{k_i} \prod_{j \in p(i,b)} \frac{1}{k_j - 1}$$

. The factor -1 in $k_j - 1$ is due to the message not returning to its point of origin. This formulation of probability of propagation of a message along a shortest path from node i to node b leads to an information theoretic formulation of the *Search Information* as

$$S(i->b) = -log_2(\sum_{p(i,b)} Pp(i,b))$$

. In other words, the search information measures the number of yes/no questions that need to be asked in order to find an object. A higher the search information means that it is more difficult to find objects in the network. At the application-level, search information provides a measure, in number of bits, of a relative size of cache or data storage that would be necessary at every node in order to

reach other nodes in the network along the shortest paths. It can be associated to the memory requirement to store routing tables to reach all nodes for a given network topology.

- **Robustness**: measures the number of edges that need to be removed in order to disrupt the network. While the edge connectivity obtained by applying a minimum cut algorithm on the network gives an indication of the minimum number that disrupt a network, it does not reflect the importance of the edge that is removed. To circumvent this issue, we introduce a measure of robustness that incrementally removes the edge of highest betweenness centrality until the network is disconnected. This measure of robustness can be understood as answering the question *how many of the most important edges need to be removed to disrupt the network?*. The algorithm corresponding to this robustness formulation is shown in Algorithm 1.

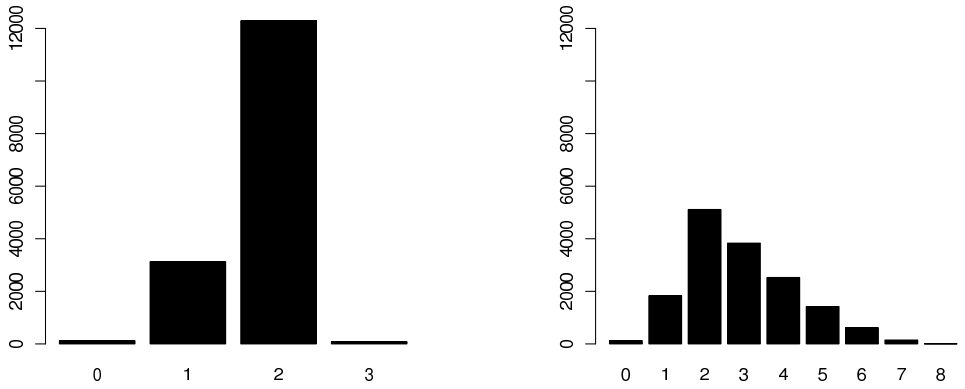
- **Clustering**: a network cluster is a densely connected group of nodes. Modifying the structure of a network affects the number and the size of clusters that can be identified. Clustering was presented in depth in Chapter 5. In this section, the number of clusters found is computed using the edge betweenness community detection presented in [63].

We now analyze the relationships between optimizing the local metrics of transitivity, assortativity, and degree network entropy on average path length, search, robustness, and clustering.

6.5.1 $G_{(125,0.2)}$ Random Graph

Table 6.4 shows the set of optimized graphs corresponding to the Random graph model, and their impact on application-level properties. The results show that all optimizations increase application-level properties as compared to the original, non-optimized

network. The most significant impact on average path length comes from maximizing assortativity, followed by maximizing degree network entropy, both modifications have the effect of increasing the homogeneity of a network neighborhood, which in the case of a random network accentuates the similarities and tends to cluster nodes and yield longer average path lengths. This interpretation is illustrated in Figure 6.7(a), that plots the shortest paths distribution for the original network and the one with optimized assortativity in Figure 6.7(b). The figures show that the shortest paths distribution stretches and flattens, with less nodes having short path lengths and many more having longer path lengths. The largest increase in search information is also obtained by maximizing the assortativity, and can be attributed to the fact that there is less *degree diversity* in the network. The robustness is most increased when the assortativity is minimized. This can be explained by the fact that as nodes connect to unlike degree nodes, the number of edges with higher betweenness increases, and it takes to remove more such edges to disrupt the network.



(a) (b)

Figure 6.7: (a) Shortest Paths Distribution for $G_{125,0.2}$ and (b) for the same network with maximized assortativity.

The effect of optimization on the network clustering is depicted in Figure 6.8. The original random graph (leftmost bar in Figure 6.8) shows three distinct homogeneous

Graph	Average Path Length	Search	Robustness
ERG	1.8 [0 %]	5.65 [0 %]	572 [0 %]
MaxTr	2.01 [11.71 %]	6.49 [14.91 %]	**539 [-5.77 %]**
MaxAs	**2.26 [25.47 %]**	**7.16 [26.82 %]**	650 [13.64 %]
MaxNtrp	2.08 [15.36 %]	5.51 [-2.42 %]	613 [7.17 %]
minTr	1.81 [0.57 %]	5.5 [-2.57 %]	870 [52.1 %]
minAs	2.03 [12.6 %]	**5.42 [-4 %]**	**735 [28.5 %]**
minNtrp	1.97 [9.14 %]	6.22 [10.1 %]	660 [15.38 %]

Table 6.4: Effect of Optimization on Global Properties for Random Graph

partitions. This is due to the homogeneity and low diameter of the random graph. Optimizations on the assortativity appear to split the network into two distinct components of equal size. As was expected minimizing transitivity is the only reconfiguration strategies that yields more diverse clusters, although not significantly in the case of the random graph, which might again be explained by the lack of *diversity* in the network.

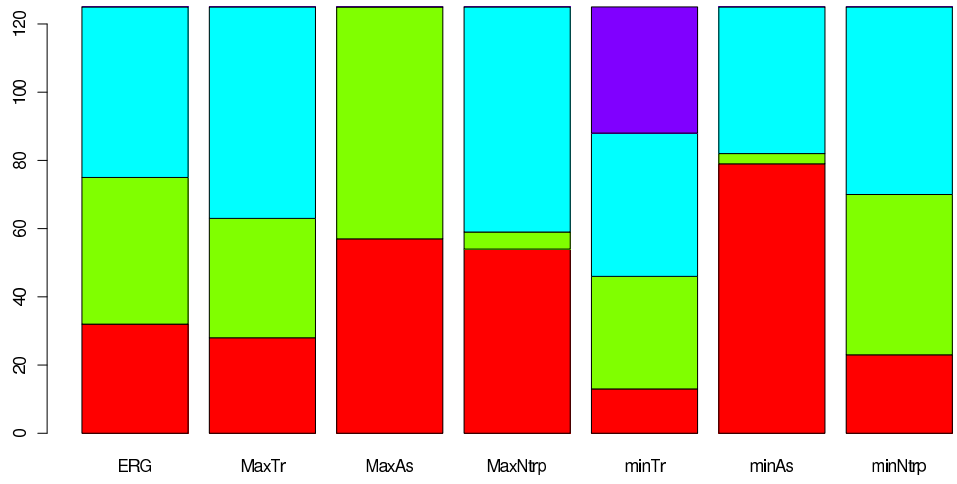
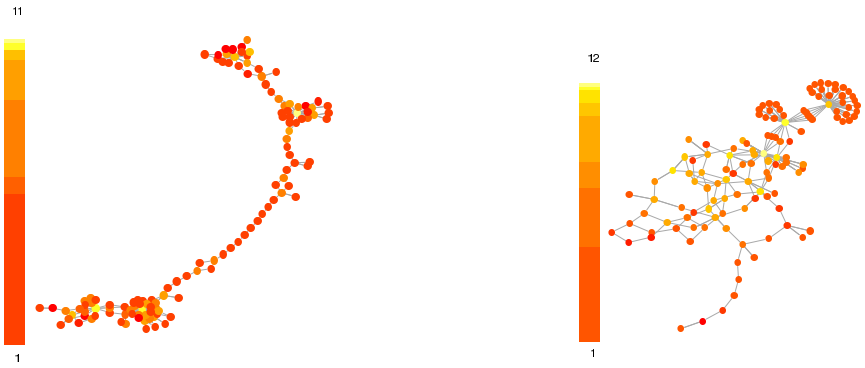


Figure 6.8: Cluster Distribution for the original and optimized random graph networks.

6.5.2 Barabási-Albert Graph

Table 6.5 presents the results of the impact of the local optimizations on the application-level properties for the scale-free power-law degree distributed network. The effect

of the optimizations on average path length reveal an increase in most cases, except when minimizing degree network entropy and maximizing assortativity. The average path length is significantly increased when the degree network entropy is maximized. The effect of maximizing degree network entropy is to create a heterogeneous neighborhood for a given node, in the scale-free network with power-law distribution, this results in a stretched out line-like topology that results in a high average path length. This interpretation is shown in Figure 6.9 using a force-directed layout representation [2] of the maximized average path lengths network and the maximized degree network entropy, on which the apparent linearity of the topology is revealed. Further, the search information is increased most when degree network entropy is maximized and decreased most when degree network entropy is minimized. The robustness remains unaffected, a result that might be attributed to the fact that the network is acyclic with no transitivity and does no have alternate paths to route messages through.



(a) (b)

Figure 6.9: (a) Maximized Average Path Length for Scale-Free network. (b) Maximized Degree Network Entropy for Scale-Free network.

The effect of optimization on the network clustering is depicted in Figure 6.10. The degree heterogeneity of the scale-free network shows less distinct clustering distributions as in the case of the random graph presented in the previous section. Notably, optimizations of the degree network entropy, decrease the number of clusters while homogenizing the clusters size. The same *diversification* factor applied by minimizing

Graph	Average Path Length	Search	Robustness
BAG	4.61 [0 %]	11.11 [0 %]	1 [0 %]
MaxTr	4.91 [6.49 %]	11.29 [1.6 %]	1 [0 %]
MaxAs	3.94 [-14.46 %]	9.65 [-13.15 %]	1 [0 %]
MaxNtrp	11.18 [142.48 %]	**14.8 [33.14 %]**	1 [0 %]
minTr	4.96 [7.64 %]	11.95 [7.54 %]	1 [0 %]
minAs	**14.05 [204.82 %]**	14.26 [28.33 %]	1 [0 %]
minNtrp	**3.78 [-18.08 %]**	**9.53 [-14.22 %]**	1 [0 %]

Table 6.5: Effect of Optimization on Global Properties for Scale-Free Graph

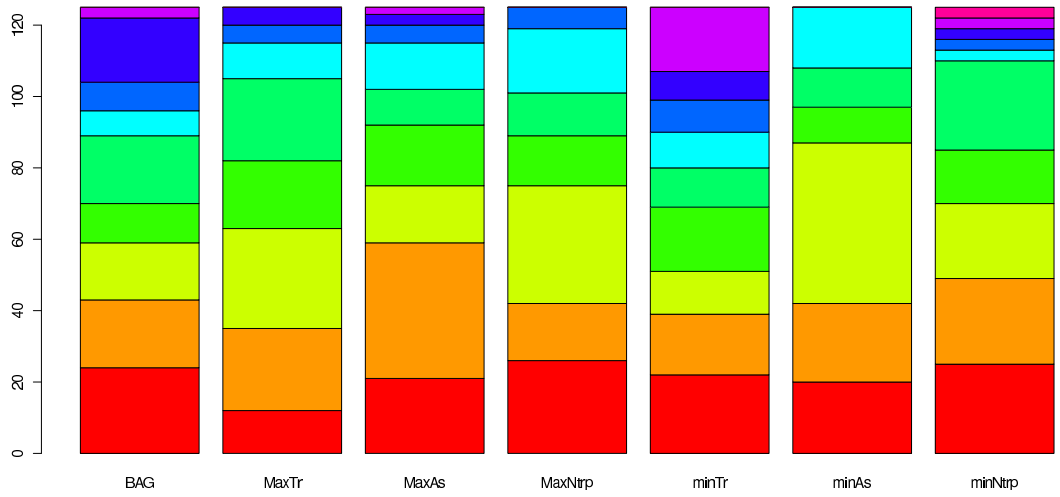


Figure 6.10: Cluster Distribution for the original and optimized scale-free networks.

assortativity yields one large cluster and a few smaller ones that result from the degree diversification imposed by the degree correlation minimization.

6.6 Case-Studies: Real-World Networks

The previous section analyzed two widely used network models and the impact of local reconfigurations on routing, search, robustness, and clustering. This section considers two real-world networks, the Gnutella Peer-to-Peer file sharing and the Canadian Autonomous System. Both networks are representative of the emergent complexity of next-generation networks. Figure 6.11(a) shows a force-base layout representation of the Autonomous System (AS) of the Canadian Internet *circa* 2007. Each node in the graph represents an Autonomous System, and a link between two Autonomous Systems is a peering relationship between two AS's. Figure 6.11(b) shows a portion of the Gnutella Peer-to-Peer network *circa* 2005. Each node in the graph represents a client/server peer that can exchange data on the network, links represent application-level connections between the peers.

(a) (b)

Figure 6.11: (a) The Autonomous System of the Canadian Internet. (b) The Gnutella P2P Network.

6.6.1 Gnutella P2P Network

The Gnutella [67] P2P file-sharing application is a popular overlay network that provides its users with the functionality to search for files by name and regular expressions on names. The search is initiated from a client interface and propagated through the network by flooding through an ad-hoc hierarchy of peers. Peers with high uptime and bandwidth can be promoted to the role of ultra-peer, acting more like a hub than an edge peer. The search is then directed towards peers that have registered ownership of the matching content with an ultra-peer.

A Gnutella network graph, circa 2005, is shown in Figure 6.11(b). The figure represents a small section of the entire network, and consists of a total of 393 peers and 471 links.

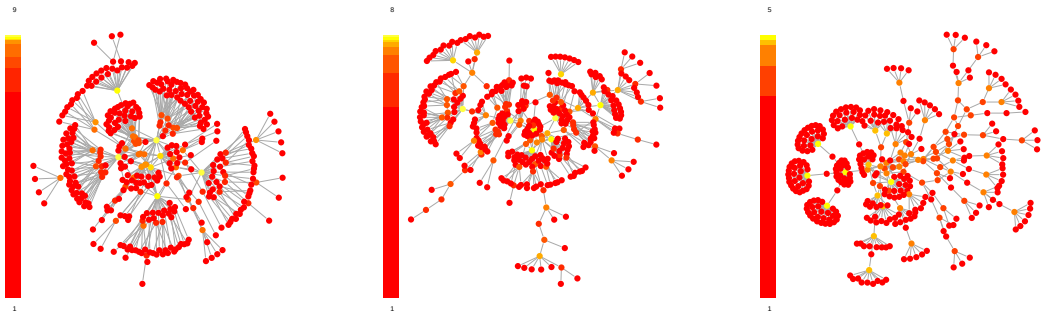


Figure 6.12: Force-based layout of Maximized Gnutella Network. (a) Maximized Assortativity. (b) Maximized Transitivity. (c) Maximized Degree Network Entropy

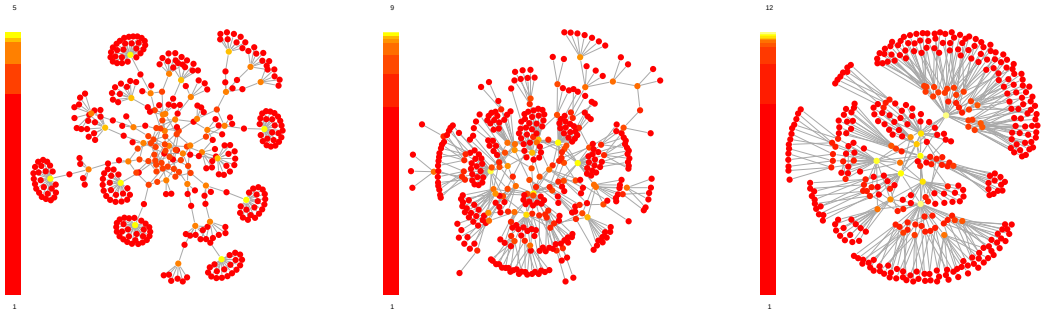


Figure 6.13: Force-based layout of Minimized Gnutella Network. (a) Minimized Assortativity. (b) Minimized Transitivity. (c) Minimized Degree Network Entropy

The results of univariate optimizations are presented in Table 6.6. The degree distribution of the network reveals that it is more similar to the scale-free network than the random network. The network transitivity of the original network is 0, an interesting result that reflects the acyclic tree structure of the peers hierarchy. The assortativity is widely optimized between $[-82.75\%, 36.75\%]$, while the degree network entropy is optimized between $[-0.82\%, 2.39\%]$, results that are comparable to the scale-free network model previously presented. Given the similar ranges of optimizations, we expect the impact of optimizations on the modeled network to be reflected on the real-world gnutella network. In particular, that the local strategy of minimizing degree network entropy results in minimizing routing and search.

Graph	**Transitivity**	**Entropy**	**Assortativity**
GNUTELLA	0 [0 %]	0.04 [0 %]	0.22 [0 %]
MaxTr	**0.04 [3.58 %]**	0.04 [0.33 %]	0.02 [-20.57 %]
MaxAs	0.02 [2.49 %]	0.04 [-0.16 %]	**0.59 [36.75 %]**
MaxNtrp	0.01 [0.87 %]	**0.06 [2.39 %]**	-0.44 [-66.25 %]
minTr	0 [0 %]	0.04 [0.31 %]	-0.03 [-25.5 %]
minAs	0 [0.33 %]	0.06 [2.36 %]	**-0.61 [-82.75 %]**
minNtrp	0.03 [3.47 %]	**0.03 [-0.82 %]**	0.15 [-6.85 %]

Table 6.6: Gnutella Network Analysis

Impact at the Application-Level:

Graph	**Average Path Length**	**Search**	**Robustness**
GNUTELLA	4.99 [0 %]	11.9 [0 %]	9 [0 %]
MaxTr	5.97 [19.46 %]	**15.21 [27.87 %]**	**9 [0 %]**
MaxAs	4.83 [-3.26 %]	**11.47 [-3.54 %]**	**1 [-88.89 %]**
MaxNtrp	**8.11 [62.47 %]**	14.2 [19.38 %]	**1 [-88.89 %]**
minTr	5.07 [1.51 %]	11.89 [-0.01 %]	6 [-33.33 %]
minAs	9.2 [84.23 %]	13.68 [15.04 %]	2 [-77.78 %]
minNtrp	**4.38 [-12.19 %]**	12.04 [1.21 %]	7 [-22.22 %]

Table 6.7: Gnutella Application-Level Properties

Table 6.7 shows that the average path length is increased when maximizing the degree network entropy and decreased when minimizing degree network entropy. This

result is identical to the modeled network and shows that for power-law degree distributed networks, the degree network entropy is a useful local strategy to guide a network towards increasing or decreasing average path length. The search is increased most upon maximizing transitivity, which also increases the average path length. This result can be explained by the fact that as a neighborhood becomes more densely connected, the number of questions to ask to find an object is also increased, this result concurs with a previous finding presented in [68]. The same justification can be applied to the increase in search information when maximizing degree network entropy. However, maximizing assortativity results in a decrease in search information, a surprising result as, although the degree homogeneity is increased in this case as well, unlike transitivity and degree network entropy the homogeneity takes into consideration the node at which the metric is being computed, leading up to a lower search information. Finally robustness is significantly decreased when maximizing assortativity and degree network entropy due to the homogenization of the network's degrees.

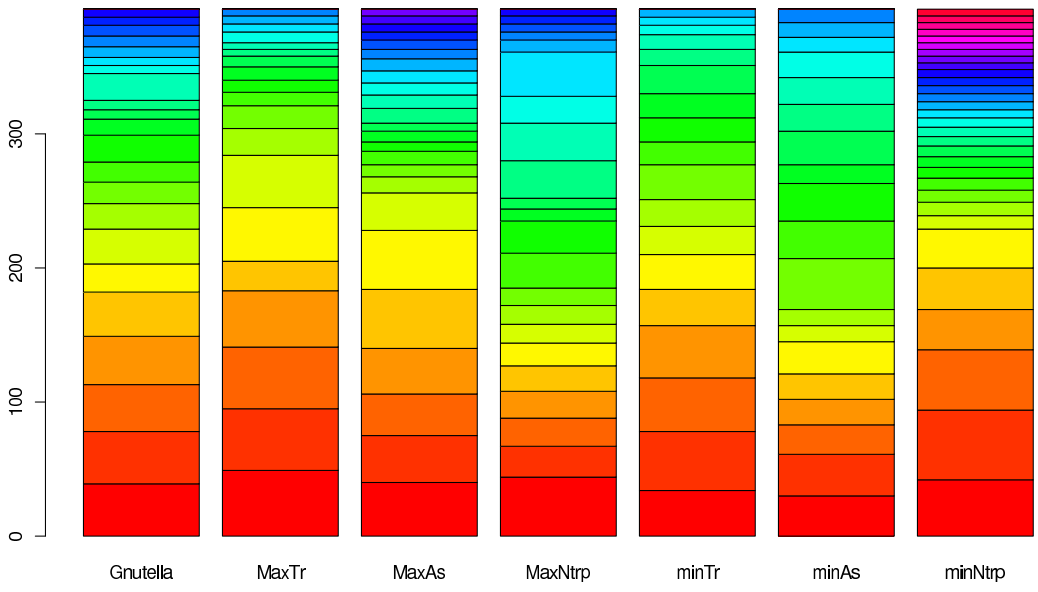


Figure 6.14: Cluster Distribution for the original and optimized Gnutella networks.

The effect of optimization on the network clustering is depicted in Figure 6.14. The impact of the local strategies on clustering shows that optimizing transitivity is the only strategy that homogenizes the number and size of clusters. The clusters identified in all strategies do not differ significantly, a result that may be attributed to the *long tail* distribution characteristic of power-law degree distributed networks, that limits the number of high degree nodes and therefore does not enable nodes to group into distinct partitions.

6.6.2 Canadian Autonomous System

The graph representation of the Canadian Autonomous System 6.11(a) circa 2007 contains 496 nodes and 814 undirected links. Results of the local optimizations and the impact on application-level properties are presented in Table 6.8 and displayed in Figure 6.15. As in previous instances of scale-free networks, the Average Path Length is decreased exclusively upon applying a reconfiguration strategy that minimizes the degree network entropy, which also has the least impact on decreasing robustness. In the case of the Canadian Autonomous System, unlike in the previous network models, search is decreased when applying a strategy that minimizes transitivity. Minimizing transitivity tends to create more tree-like networks that, for a given degree distribution such as the Canadian AS network, may result in lower search costs. All optimization strategies yield a decrease in the measure of robustness, revealing a trade-off between improving the measures of average path length, and search, at the expense of decreased robustness.

The effect of optimization on the network clustering is depicted in Figure 6.17. The impact of the local strategies on clustering shows that optimizing degree network entropy and minimizing assortativity are the only strategies that homogenize the number and size of clusters. While minimizing degree network entropy *diversifies* the degree neighborhood of a node, minimizing assortativity yields a neighborhood that has dissimilar neighborhood degree than the node considered. Yet, these contrasting results

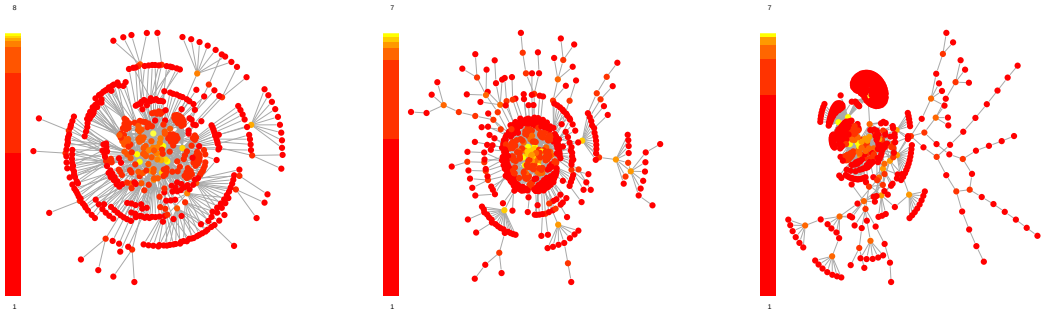


Figure 6.15: Force-based layout of Maximized Canadian Autonomous System Networks. (a) Maximized Assortativity. (b) Maximized Transitivity. (c) Maximized Degree network Entropy

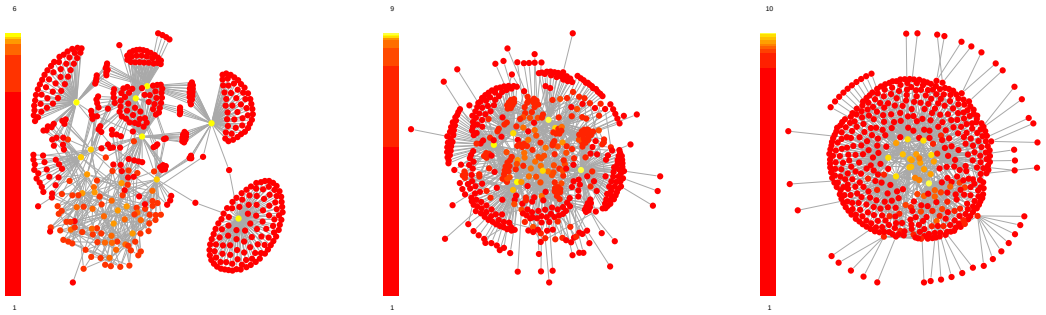


Figure 6.16: Force-based layout of minimized Canadian Autonomous System Networks. (a) Minimized Assortativity. (b) Minimized Transitivity. (c) Minimized Degree Network Entropy

Graph	Transitivity	Entropy	Assortativity
CANADA07	0.03 [0 %]	0.07 [0 %]	-0.22 [0 %]
MaxTr	**0.09 [6.67 %]**	0.07 [0.38 %]	-0.07 [15.55 %]
MaxAs	0.07 [4.23 %]	0.07 [0.23 %]	**0.1 [31.8 %]**
MaxNtrp	0.02 [-0.51 %]	**0.09 [2.48 %]**	-0.38 [-16.17 %]
minTr	**0 [-2.62 %]**	0.06 [-0.21 %]	-0.33 [-10.83 %]
minAs	0 [-2.56 %]	0.09 [2.29 %]	**-0.56 [-33.74 %]**
minNtrp	0.06 [3.86 %]	**0.04 [-2.81 %]**	-0.14 [8.02 %]

Graph	Average Path Length	Search	Robustness
CANADA07	3.25 [0 %]	10.6 [0 %]	47 [0 %]
MaxTr	4.35 [34 %]	12.26 [15.7 %]	**1 [-97.87 %]**
MaxAs	3.69 [13.58 %]	11.18 [5.5 %]	**1 [-97.87 %]**
MaxNtrp	**5.36 [65.15 %]**	**12.38 [16.84 %]**	2 [-95.74 %]
minTr	3.26 [0.32 %]	**9.84 [-7.12 %]**	31 [-34.04 %]
minAs	4.19 [28.94 %]	10.79 [1.85 %]	2 [-95.74 %]
minNtrp	**2.96 [-8.97 %]**	11.16 [5.36 %]	22 [-53.19 %]

Table 6.8: Results of Local and Application-Level Network Optimizations for the Canadian Autonomous System circa 2007

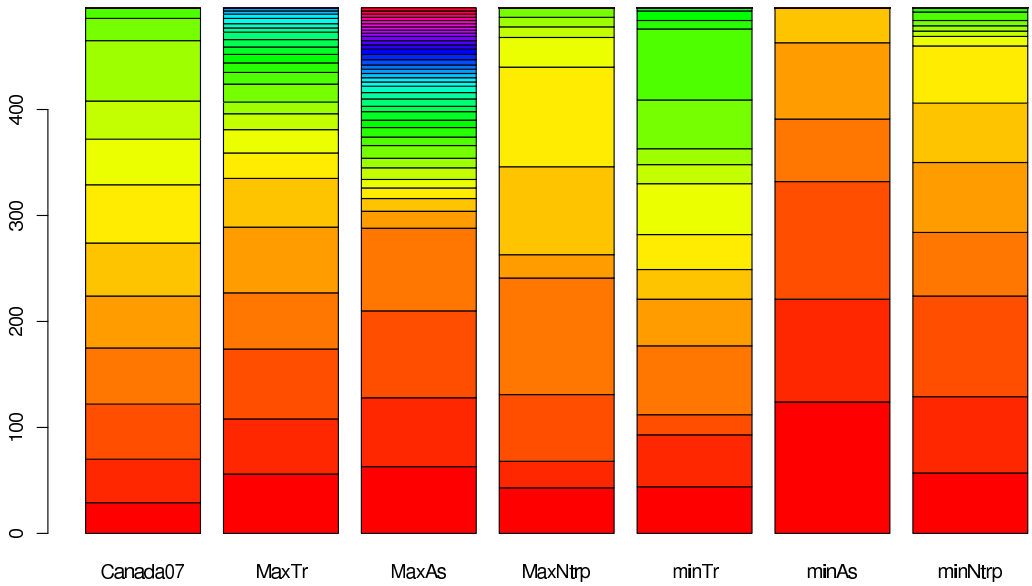


Figure 6.17: Cluster Distribution for the original and optimized Canadian Autonomous System networks.

yield more homogeneous partitioning of the network, and reflect the importance that an initial degree distribution plays in determining application-level properties of a network.

6.7 Summary

The topology of emerging networks is the result of local adaptations that are driven by administrative policy and engineering decisions. At a large scale, this decentralized evolving process yields structural properties that need to be better understood. While understanding the impact of such local to global properties has received most attention in the areas of statistical physics and mathematics, it remains absent from the network engineering research. This chapter presented three local metrics deemed representative of local structural characteristics of a node and its network neighborhood. Using simulated annealing optimization, the network was modified based on adaptations of the local metrics, and the impact on the application-level functionalities of path length, search, robustness and clustering was discussed. The results show that local perturbations that increase the degree network entropy, i.e., increase the degree heterogeneity of a node's neighborhood, also increase the average path length, and has an impact on search, robustness, and clustering that is topology dependent. Minimizing the degree network entropy, i.e., increasing the degree homogeneity of a node's neighborhood, decreases the average path length for the two real-world networks considered. This result suggests that for such topologies, minimizing the degree network entropy can be used as a local strategy towards decreasing overall average path length and search at the cost of decreased network robustness.

Chapter 7
Conclusions and Future Work

We shall not cease from explorations
And the end of all our exploring
Will be to arrive where we started
And know the place for the first Time

- T.S. Eliot

7.1 Conclusions

Research on emerging networks is facing an unusual development, to the traditional top-down design problems is now added the challenge of understanding the bottom-up processes that can explain the observed structures of large and dynamic networks. Understanding the decentralized evolution of computer networks such as the Internet has been added to a list of open problems in complexity theory which have previously been observed in economics, social science, mathematics, and physics. In particular, understanding the evolution of large-scale computer network topologies, that define the structural properties of networks, is becoming increasingly important due to the increasing number of overlay networks being deployed. These virtualized networks offer software-level configurations that occur on top of the physical topology, and exhibit frequent reconfigurations and high dynamicity. The evolution of the structure of networks follows a process in which nodes and links are added, deleted, and reconfigured dynamically. Exploring the resulting structures of such evolutions is a combinatorial

problem that is often computationally intractable.

Current efforts in network topology research have proposed topology awareness to augment information at each node with topological information to improve the local decision making. However much of the global impact on the network topology remains absent of these studies.

Network topology modeling attempts to isolate and replicate the generating principles that drive the evolution of real-world networks. These approaches are based on collecting real-world network data and quantifying the model by matching metrics between the real-world and the modeled networks. Over the years, new metrics introduced have helped refine models by adding more and more accuracy into identifying better models. However, the number of metrics proposed continues to grow.

Emergent properties and self-organization have been applied to topologies to evolve the network based on adaptive and local information alone. Despite the urgent need to better understand the impact of such local ad-hoc evolving processes at the application level, such as routing, search, robustness, and clustering, this branch of research has received most attention from the statistical physics and biological sciences literature but have been absent from the computer network research.

The material presented in this thesis falls in the context of emergent topologies. The objective of this research is to (1) identify local canonical topological metrics, (2) apply adaptive strategies based on these metrics to modify various network topologies, and finally (3) analyze the networks resulting from these adaptations for the application-level properties of routing, search, robustness, and clustering.

The three local metrics identified, transitivity, assortativity, and entropy, are presented as canonical properties of the topology relating respectively to aspects of resilience, homogeneity, and information. The approach consists in optimizing graph theoretic representations of network topologies using simulations, and quantitatively

analyzing the impact of these optimizations at the application-level for routing, search, robustness, and clustering.

The results show that local perturbations that increase the degree entropy, i.e., increase the degree heterogeneity of a node's neighborhood, also increase the average path length, and has an impact on search, robustness, and clustering that is topology dependent. A key outcome of this thesis is the identification of network entropy minimization as a useful local rewiring strategy to decrease average path length and search cost, while homogenizing the size of network clusters and having a low impact on robustness when applied to power-law degree distributed networks that prevail in real-world networks.

7.2 Prospects and Future Work

The exploratory and empirical nature of this research leaves much to study and explore as future work. Through this research, significantly more questions have been asked than answers found. The nature of such complex problems requires an inductive approach and intuitive understanding of the processes that drive the evolution of such systems.

In particular,

- While the local strategy of minimizing network entropy was shown to decrease average path length and search in scale-free networks, the identification of new local metrics and their relation to application-level properties remains an open problem that requires further investigation. Applying the same methodology on different metrics could further the understanding of the driving forces that control large complex networks such as the Internet.

- Local metrics based on local properties other than degree need to be evaluated.

For example, context information such as location or characteristics of every node.

- In this thesis, all edges of the considered networks were assumed to have constant unit cost. Addressing the problem for edges with arbitrary distributions and the integration of a network flow formulation to the existing structural properties remains a challenging open problem.

- Building a mapping between the distribution of network properties and observed global structures provides heuristics that can improve on current ad-hoc approaches, and in particular on understanding the structure of large-scale networks. For example, such heuristics could be used to improve predictions on the complex non-linear evolution of the Internet Autonomous Systems that is driven by political and engineering decisions.

- The dynamics of real-world large-scale networks such as the Internet collected by agencies such as CAIDA [13] should be used to model strategies, other than structural, that drive the evolutionary process.

- As a result of the advances in understanding such complex emergent properties, the development of a suite of software recommender tools to assist Autonomous Systems network administrators in the reconfigurations of Autonomous Systems should be undertaken. Such tools would be useful to optimize the reconfigurations driven by administrative policies and technical engineering decisions while optimizing application-level properties.

Metric Name	**Description**	**Equation/Symbol**	**Bounds**	**Result Size**
Number of Nodes	number of interacting nodes in the connected graph	$V(G)$	$[1, n]$	1
Number of Edges	number of edges in the connected graph	$E(G)$	$[(n-1), \frac{n(n-1)}{2}]$	1
Average Degree	average number of neighbors per node over all nodes	$\frac{1}{n}\sum_{i=1}^{n} d_i$	$[\frac{2(n-1)}{n}, n-1]$	1
Diameter	longest path between any two nodes	$C_D = max_{i,j \in V} \delta(i,j)$	$[1, n-1]$	1
Assortativity	measures the preference of a node to connect to like degrees.	$r = \frac{M^{-1}\sum_i j_i k_i - [M^{-1}\sum_i \frac{1}{2}(j_i + k_i)]^2}{M^{-1}\sum_i \frac{1}{2}(j_i^2 + k_i^2) - [M^{-1}\sum_i \frac{1}{2}(j_i + k_i)]^2}$	$[-1, 1]*$	1
Degree Network Entropy	Expected Self-Information based on a node degree	$C_{ne} = -\sum_{i \in V(G)} p_i log(p_i), p_i = \frac{d_i}{\sum_i d_i}$	$[\frac{2logn}{n-1}, log(\frac{n(n-1)}{2})]$	1

Table 1: Network Metrics. (*) indicates normalized values. M is the number of edges. d_i the degree of node i. $\delta(i,j)$ the distance between nodes i and j.

Metric Name	Description	Equation/Symbol	Bounds	Result Size
Degree	the number of adjacent nodes, i.e., neighbors, of a node in the graph.	$d_i, i \in V(G)$	$[1, n-1]$	n
Betweenness Centrality	number of times a node is present in the shortest pathσ between all pair of nodes s, t.	$C_B(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$	$[0, 1]*$	n
Spectrum	set of eigenvalues of the adjacency matrix of the graph.	$p_A(\lambda) = det(A - \lambda I)$	$[-, n-1]$	n
Eigenvector Centrality	eigenvector corresponding to the first eigenvalue.	$x_i = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j$	$[-1, 1]$	n
Transitivity	measures the probability that neighbors of a node are themselves neighbors. Computed as the number of triangles divided by number of triples.	$C_i = \frac{\lvert\{e_{jk}\}\rvert}{d_i(d_i-1)} : v_j, v_k \in N_i, e_{jk} \in E.$	$[0, 1]$	n
Degree Distribution	frequency of appearance of each unique degree d_i of the set of represented degrees in the graph.	$p(k) = \frac{1}{n} \sum_{v \in V \mid deg(v)=k} 1$	$[0, 1]$	d
Joint Degree Distribution	distribution of average neighbor degree of a node of average degree d.	$p(k_1 k_2) = \frac{\mu_{k_1,k_2} M_{k_1,k_2}}{2M}$ $\mu_{k_1,k_2} = \begin{cases} 1 & \text{if } k_1 = k_2 \\ 0 & \text{otherwise} \end{cases}$	$[0, 1]$	d

Table 2: Network Metrics continued. (*) indicates normalized values. M is the number of edges. M_{k_1,k_2} is the number of edges between all nodes of degree k_1 and k_2. $\sigma_{s,t}$ is the total number of shortest paths between node s and node t. $\sigma_{s,t}(v)$ is the number of shortest paths between nodes s and t that go through node v. A is the adjacency matrix representation of the graph.d_i degree of node i. N_i the set of nodes in the neighborhood of node i. e_{jk} the number of edges connecting all nodes in the neighborhood of node i.

Appendix A

Glossary

A.1 Graph Theory

- **Subgraph** : all edges and vertices of a subgraph are included in the supergraph
- **Adjacency Matrix** : nodes in rows and columns.

$$a_{i,j} = \begin{cases} 1 & \text{if (i,j) } \varepsilon \text{ E} \\ 0 & \text{otherwise} \end{cases}$$

- **Incidence Matrix (for a DAG)** : rows are the edges and columns the nodes (edge-node incidence matrix), or rows are the nodes and columns the edges (node-edge incidence matrix).

$$b_{i,j} = \begin{cases} -1 & \text{if edge} x_j \text{ leaves vertex i} \\ 1 & \text{if edge } x_j \text{ enters vertex i} \\ 0 & \text{otherwise} \end{cases}$$

- **Admittance (Laplacian) Matrix of a Graph** : is always positive semi-definite, so all its eigenvalues are non-negative $x^T Lx = \sum_{(u,v)\varepsilon E}(x_u - x_v)^2$. Particularly its second eigenvalue is strictly positive. **L=D-A** where D is the degree matrix and A is the adjacency matrix.

$$l_{i,j} = \begin{cases} -1 & \text{if (i,j) } \varepsilon \text{ E} \\ d_i & \text{if i=j, and} \\ 0 & \text{otherwise} \end{cases}$$

- **Degree** : Number of edges incident with a vertex in an undirected graph. In digraphs, distinguish indegree and outdegree
- **Regular of Degree r** : every vertex has the same degree r
- **Walk** : any sequence of consecutive edges
- **Path/Trail/Open Walk** : vertex set $x_1, ..., x_n$ with edges x_i, x_{i+1}
- **Hamiltonian Path** : path of an Undirected graph that visits every vertex exactly once
- **Eulerian Path** : path that visits every edge exactly once
- **Cycle/Circuit/Closed Walk** : vertex set $x_1, ..., x_n$ with edges x_i, x_{i+1} and x_n, x_1
- **Bridge/Cutpoint** : if edge is cut / vertex removed, the number of components is increased
- **Girth** : length of shortest simple cycle
- **Circumference** : length of longest simple cycle
- **Diameter** : largest distance between the vertices
- **Betweenness** : Measures the number of times a node on shortest paths between all pairs of vertices. Betweenness is a measure of centrality of the network, as the higher it is the more central is.
- **Properly Colored** : if each vertex is colored so that adjacent vertices have different colors
- **K Colorable** : if it can be properly colored using k colors
- **Bipartite Graph** : the vertices of a bipartite graph can be divided into two disjoint sets, for which there is no edge between any two vertices of a same set.

- **Chromatic Number k** : if graph is k colorable
- $\mathbf{K_n}$: the complete graph on n vertices
- **Tree** :a graph in which any two vertices are connected by exactly one path.
- **Forest** : a graph in which any two vertices are connected by at most one path. Disjoint union of trees
- **Embedding** : drawing of the representation of a graph on any surface such that no edges intersect.
- **Edge Connectivity** : the minimum number of edges of a connected graph whose removal decreases the rank of the graph by one.
- **Vertex Connectivity** : the minimum number of vertices of a connected graph whose removal leaves the graph disconnected.
- **Separabale Graph** : if the vertex connectivity is one
- **Line Graph** : vertices correspond to the edges with two vertices being adjacent if and only if the corresponding edges in G have a vertex in common
- **Planar Graph** : graph that can be embedded in a plane so that no edges intersect (Kuratowski's theorem, finite graph is planar if none of its subgraph is an expansion of $K_5 or K_{3,3}$. Any graph with number of edges e ¿ 3n-6 is Nonplanar.
- **Dual Graph** : exists only if graph is planar. Each vertex of G* corresponds to a face of G. Each edge of G* crosses the edge of G that connects the two vertices between the two faces of G. Therefore G* has as many vertices as G has faces, as many edges as G, and as many faces as G has vertices.
- **Ramsey Theorem** : In a complete graph of order $R(n_1, ..., n_c; c)$with n vertices, if the edges of the graph are colored using c colors, then there is a complete subgraph n_i of color i.

- **Number of Connected Components** : is given by the multiplicity of the eigenvalue 0 of the Laplacian of the graph.

- **Fiedler value/ Second eigenvalue of the Laplacian matrix of a Graph** : If a symmetric matrix is positive semi definite then all its eigenvalues are non-negative. The multiplicity of the smallest eigenvalue 0 gives the number of connected components. For a connected graph, the dimension of the eigenspace of 0 is 1 and therefore the magnitude of the second eigenvalue (Fiedler value) is a measure of how well connected the graph is. For a planar graph $\lambda_2 \leq \frac{8d}{n}$.

- **Expander Graph** : a graph in which subset of vertices have high vertex or edge expansion to the complementary set of vertices. The high connectivity between the various sets acting as redundancy paths might explain why it is used in Error Correction code. (Low Density Parity Check)

- **Spanning Tree** : Tree which includes every vertex of a graph

- **Minimum Spanning Tree** : Find the tree that connects all vertices while minimizing the sum of the weight edges.

- **Steiner Tree Problem** : Find the shortest network that spans a given set of points, given that new points can be added to the network at any time. (MST is a Steiner Tree problem with a fixed given set of points). The STP is an NP-complete problem.

- **Kirchhoff's Theorem** : Given a connected graph G with n vertices, let $\lambda_1, \lambda_2, ..., \lambda_{n-1}$ be the non-zero eigenvalues of the admittance matrix of G. Then the number of spanning trees of G is $G = \frac{1}{n} * (\lambda_1 \lambda_2 ... \lambda_{n-1})$

- **Cayley's Formula** : if n is a positive integer, the number of trees on n labeled vertices is n^{n-2}

- **Genus of a Graph** : the minimum number of handles that must be added to the plane to embed the graph without any crossings. A planar graph therefore has graph genus 0.

- **Clustering Coefficient** : the number of links between the neighbors of a node divided by the total number of links that could exist. It is a measure of how well connected a neighborhood is.

- **Dominating set** for a graph G=(V,E) is a subset V' of V such that every vertex not in V' is joined to at least one member of V' by some edge.

- **Domination number** of G is the number of vertices in the smallest dominating set for G. partition the vertices of a graph into a given number of dominating sets; the maximum number of sets in any such partition is the **domatic number** of the graph.

- **Unit Disk Graph** In geometric graph theory , a unit disk graph is the intersection graph of a family of unit circles in the Euclidean plane. That is, we form a vertex for each circle, and connect two vertices by an edge whenever the corresponding circles cross each other.

A.2 Miscellaneous Principles and Concepts

- **Pareto Principle (80-20 rule)**: for many phenomena 80% of consequences stem from 20% of the causes.

- **Small-world Network**: Network in which any node can reach any other node in a small number of steps. This type of network can be established by analyzing the network's clustering coefficient and average path length. If clustering coefficient is larger than normal and average path length is smaller than normal, then it is likely to be a small-world network.

- **Entropy (Concept)**: Let's consider an element in a given observable state *s*; Entropy is a measure of the number of possible low-level states *s'* that manifest the system *s*. A system is said to be ordered when it has few possible configurations, therefore the number of low-level configurations that is linked to entropy is also linked to order. Entropy is a measure of disorder of a system. There is no single entropy across sciences, thermodynamics, statistical mechanics, information theory, each have their own definition of entropy that are however linked to each other.

- **Self-Information / Surprisal**: The amount of knowledge about the outcome of an event that adds to global knowledge. Self-Information is also called Surprisal because it is a measure of the surprise factor that the realization of an event induces. The more likely an event is, the less surprised one is. The occurrence of an event, A, is measured by its probability of occurrence p(A); the unit of self-information is binary digit (bit), the probability of occurrence is therefore expressed in number of bits, or $log_2(\frac{1}{p(A)})$.

- **Information Entropy**: Introduced by Claude Shannon to measure the amount of randomness in a signal, information entropy is expressed as the expected value of the self-information (surprisal), $H(x) = \sum i = 1np(x_i)log_2(\frac{1}{p(x_i)})$.

- **Soft Computing**: An approach to solving problems that is inspired from human reasoning, as opposed to rigid computer logic. This form of computing is embodied by techniques such as Fuzzy Logic, Neural Networks, Probabilistic Reasoning (such as genetic algorithms, Bayesian networks).

- **Ergodicity**: An ensemble is ergodic if a small subset of the ensemble at time *t* "behaves" the same as the average of the whole over time. For example the ensemble of articles published in a newspaper is ergodic: the number of errors in the entire newspaper equate the number of errors an editor makes over time.

- **Markov chain**: the transition probability to state *(n+1)* is determined only by the current state *(n)* and not by the whole history. Therefore, a process modeled by a Markov Chain is also known to be memoryless.

- **Ergodic Markov Process**: Connotes reachability of any pair of states in the transition graph. Translates into a strongly connected transition matrix, i.e. there is a non-zero transition probability for any (i,j) in the transition matrix.

- **Regular Markov Process**: There is a sequence of edges of length exactly equal to *k* between any pair of vertices in the transition graph.

- **Periodic Markov Process**: An ergodic process is periodic if it can enter a state only at specific periodic intervals.

A.3 Probability and Statistics

- **Z-score**: the number of standard deviations the value is from the sample mean of the data set. That is,

  $z - score = \frac{x_i - \bar{(}x)}{s}$

A.4 Complexity classes

- **P**: decision problems that can be answered by a deterministic machine in Polynomial time.

- **NP**: decision problems that can be solved by a Non-deterministic machine in Polynomial time. Another way to put it is that it's a type of problem for which a solution can be verified but not established in polynomial time on a deterministic machine.

- **NP-Hard**: A problem L in NP is reducible to another problem not necessarily in NP. NP-Hard problems are "Harder" than NP-complete in the sense that

the problems reduce to a set of problems that might not even be decidable, and therefore not even in NP.

- **NP-Complete**: A problem L in NP is polynomial-time many-one reducible to another problem in NP (hence the difference with NP-Hard). Reduces to a decidable problem.

- **Kolmogorov-Smirnov test** (often called the K-S test) is used to determine whether two underlying probability distributions differ, or whether an underlying probability distribution differs from a hypothesized distribution, in either case based on finite samples.

A.5 Evolutionary Computing

- **Genetic Algorithms**: The candidate solutions (chromosomes in GA) are represented as a string of symbols or numbers (often in binary). The primary genetic operator is recombination but selection and mutation are used to maintain diversity.

- **Genetic Programming**: The candidate solutions are represented as computer programs, their fitness is evaluated by their ability to solve a computation problem. It uses recombination as a primary genetic operator.

- **Evolution Strategy and Evolutionary Programming**: Works with vectors of real numbers as representations of solutions, and typically uses self-adaptive mutation rates.

A.6 Misc Math concepts

- **Injective(One-to-one)** : Every element A in X has a corresponding map in codomain Y

- **Surjective (Onto)** : Every element B in codomain Y has at least an element in X mapping to it.
- **Bijective (One-to-one and Onto)** : Every element A in X maps to a single element B in Y
- **Continuous (Topological)** : Every point A belonging to a domain U of X maps to an element in V=f(U), no matter how small V is there is a small U containing A that maps inside V.
- **Eigenvalue and Eigenvector**: A matrix is a linear operator on a set of vectors, and its eigenvalues measure the scaling factor by which the set of vectors (eigenvectors) is transformed.
- **Homeomorphism/Topological Isomorphism** : 1) f is a bijection 2) f is continuous 3) the inverse function f -1 is continuous
- **Homomorphism**: a map from one algebraic structure to another of the same type that preserves all the relevant structure. For example $f(x) = 3x$ is a homomorphism but $f(x) = 3x^2$ isn't.
- **Isomorphism**: a bijective homomorphism (i.e. structure-preserving mapping).
- **Jordan Curve** : A simple (injective mapping) closed curve
- **Hausdorff Space (Housed Off)** : Space is partitioned into neighborhoods, x and y are distinct points, neighborhood U of x and V of y are disjoint
- **Cantor set (Fractals)** : The set of elements obtained from recursively removing the middle-third of the [0,1] interval. It is proved to be uncountable despite the geometric sum of the rest being the length of the set (1).
- **Lebesgue measure** : Volume in Euclidian space
- **Algebraic Topology** : Using Abstract algebra to solve topology problems

- **Open Set** : Is not opposed to ”closed”. Is open if elements can be wiggled around and still be true (e.g. 0¡x¡1 is open 0¡x= 1 is not)

- **Connected Space** : space that cannot be divided into two disjoint nonempty open sets whose union is the entire space.

A.7 Routing

- **Distance-Vector Routing** : Consider a network of nodes, each node builds its reachability routing table by collecting distances (e.g. number of hops) to its direct neighbors. Every node then shares this information with all its neighbors, so that each can build a global routing table on their own. Susceptible to count to infinity.

- **Link-State Routing** : As opposed to DV routing, each node in LS routing floods the network with information about its immediate neighbors only. The outcome of all nodes flooding all the network is that all nodes eventually have a complete view of the network. Each node then applies a shortest path algorithm on the graph of the network obtained to determine the shortest path to any other node.

- **Differences between DV and LS** : The main difference between the two routing protocols is that in DV, nodes share all the information they obtain whereas in LS nodes only share information about their immediate neighbors.

- **Wormhole routing** is a system of simple routing in computer networking based on known fixed links, typically with a short address

- **Route Flapping** occurs when a router alternately advertises a destination network first via one route then another (or as unavailable, and then available again in quick sequence).

A.8 Emergence and Self-Organization

- **Self-Organization Definition**: internal organization of an open system increases automatically without being guided by an external source.
- **Emergence**: the formation of complex patterns from simpler rules. An emergent behavior is likely to occur in a system where the number of interactions taking place between components increases exponentially with the number of components. For example, the evolving formation of the human brain. The emergent property is often unpredictable (to us humans) and unprecedented. Emergent systems appear to defy the ever-increasing entropy law, but do not violate it, as an open system can decrease its entropy while the global system's entropy increases. Examples of Emergence are:
  - Ant colonies
  - Piles of termites
  - Swarms of bees
  - Flocks of birds
  - Schools of fish
  - Herds of mammals
  - Games such as poker
  - Stock Market
  - Galaxies formation
  - Weather phenomenons such as hurricanes
  - open-source projects
  - Cities formation (with self-organization)
  - In physics, emergence does not equate with complexity but refers to the microscopic laws on top of which macroscopic laws emerge

- **Reductionism**: the nature of complex phenomenons can be reduced to the nature of its simpler parts, thereby explaining the phenomenon. Forms of reductionism are: Ontological, Linguistic, Methodological (see Occam's Razor), Linguistic, Analytical, Scientific, or Theoretical.

- **Occam's Razor**: In a nutshell, of all possible explanations of a phenomenon, the least complex one is most likely to be the correct one.

- **Holism**: In contrast to reductionism, the whole is thought to be created than the sum of its parts.

- **Teleology**: the belief and philosophical investigation on the idea that nature has a purpose. While science investigates natural laws, teleology questions the existence of an organizing principle behind these laws. For example, in teleology, a man sees because he has eyes, and has eyes so that he can see. Plato summarizes the essential idea in Phaedo as follows: ""Imagine not being able to distinguish the real cause from that without which the cause would not be able to act as a cause.". Teleological concepts heavily rely on the notion of final cause or purpose of living things. There are two essential finality concepts: extrinsic, bettering the environment, and intrinsic, bettering the self according to what is good for it. Aristotle in support of Teleology said: "Nature adapts the organ to the function, and not the function to the organ".

- **Philosophical Naturalism**: Contrasts with Teleology. In Philosophical naturalism man sees because he has eyes; it is not interested in understanding why a many has eyes. The organ serves the function. Lucretius in De Natura Rerum says: : "Nothing in the body is made in order that we may use it. What happens to exist is the cause of its use".

- **Complex Systems**: System of many parts coupled in a non-linear fashion. In non-linear systems, the system is greater than the sum of its parts. Most biological systems are complex while most engineered systems are linear. Some characteristics of a complex system:
  - A complex system is a highly structured system, which shows structure with variations
  - A complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve
  - A complex system is one that by design or function or both is difficult to understand and verify
  - A complex system is one in which there are multiple interactions between many different components
  - Complex systems are systems in process that constantly evolve and unfold over time
- **Spontaneous Order**: Emergence of order out of a chaotic context by balance of forces or natural selection. For example markets and languages.
- **Chaos Theory / Nonlinear Dynamics**: a system that exhibits sensitivity to initial conditions, where it appears to behave in random manner, even though there are no random variables and the system is deterministic.Examples are: weather, solar system, economies, population growth...
- **Self-Organized Criticality**: claims that whenever a self-organizing dynamical system is open or dissipative, it exhibits critical (scale-invariant) behavior similar to that displayed by static systems undergoing a second-order phase transition.

For example: Avalanches, Forest-fires, Sandpile, traffic jams, size of cities, size of companies, electricity blackouts.

A.9 Optimization

- **Combinatorial Optimization**: The entire solution space of a problem can be defined as the total number of permutations of all the elements that compose the system. This number gets very large very quickly. It can be illustrated as searching for the optimal path in a tree that describes the solution space. Meta-heuristic ("Beyond" "to find") algorithms are usually applied to solving combinatorial optimization problems, for example local search, simulated annealing, genetic algorithms, tabu search, ant colony optimization, GRASP (greedy randomized adaptive search procedure) , or particle swarm optimization.

- **No Free Lunch Theorem**: When averaged over all possible cost functions, all algorithms that search for an extrema of a cost function perform exactly the same. In other words, when addressing an optimization problem, the problem domain has to be very well understood to apply the most appropriate optimization algorithm as opposed to generic metaheuristic ones.

- **Simulated Annealing**: Comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. In the simulated annealing method, each point s of the search space is compared to a state of some physical system, and the function E(s) to be minimized is interpreted as the internal energy of the system in that state.

- **Tabu search**: Similar to simulated annealing, in that both traverse the solution space by testing neighbors of an individual solution. While simulated annealing generates only one neighboring solution, tabu search generates many solutions and moves to the best solution of those generated. In order to prevent cycling and encourage greater movement through the solution space, a tabu list is maintained of partial or complete solutions. It is forbidden to move to a solution that contains elements of the tabu list, which is updated as the solution traverses the solution space.

- **Ant Colony Optimization (ACO)**: Mimics ants social works such as foraging, nest building, cemetery ordering. Ants use stigmergy that exhibit the global behavior based on ants local behaviors in the absence of centralized control. The medium of communication is a pheromone deposit on the path taken by an ant. Ants follow a reinforced pheromone trail. Using this principle, foraging enables ants to find the shortest path to food from the nest.

- **Particle Swarm Optimization (PSO)**: Is a swarm intelligence mechanism. PSO are based on a multi-dimensional space in which particles (agents) navigate freely given a position and a velocity. The next position of a particle is determined by a reinforced signal emitted by other particles in the population. In this sense an ACO can be categorized as a type of PSO.

- **Genetic Algorithms**: *see Genetic Algorithms in Evolutionary Computing section*

- In game theory, the **Nash equilibrium** (named after John Forbes Nash, who proposed it) is a kind of solution concept of a game involving two or more players, where no player has anything to gain by changing only his or her own strategy unilaterally. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a

Nash equilibrium. The concept of the Nash equilibrium (NE) is not exactly original to Nash (e.g., Antoine Augustin Cournot showed how to find what we now call the Nash equilibrium of the Cournot duopoly game). Consequently, some authors refer to it as a Nash-Cournot equilibrium. However, Nash showed for the first time in his dissertation, Non-cooperative games (1950), that Nash equilibrium must exist for all finite games with any number of players. Until Nash, this had only been proved for 2-player zero-sum games by John Von Neumann and Oskar Morgenstern (1947).

References

[1] Citeseer, "Computer and information science papers citeseer publications researchindex." url: http://citeseer.ist.psu.edu/, 2007.

[2] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett*, vol. 31, pp. 7–15, 1989.

[3] J. H. Miller and S. E. Page, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, 2007.

[4] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, pp. 4–10, IEEE Computer Society, 2004.

[5] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review, cond-mat/0303516*, vol. 45, pp. 167–256, Mar. 2003.

[6] G. Huston, "The bgp report," 2006.

[7] K. P. Gummadi, R. Gummadi, S. D. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of dht routing geometry on resilience and proximity," in *Proceedings of the ACM SIGCOMM 2003*, 2003.

[8] J. M. Carlson and J. Doyle, "Highly optimized tolerance: A mechanism for power laws in designed systems," *Physics Review E*, vol. 60, pp. 1412–1427, 1999.

[9] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, pp. 131–145, ACM Press, 2001.

[10] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *SIGCOMM*, pp. 251–262, 1999.

[11] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, vol. 38, pp. 34–41, 2005.

[12] D. Talbot, "The internet is broken," *Technology Review*, December 2006.

[13] CAIDA, "Cooperative association for internet data analysis." url: http://www.caida.org, 2006.

[14] ISC, "Internet systems consortium." url: http://www.isc.org, 2006.

[15] S. Dorogovtsev and J. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, 2004.

[16] K. Sneppen, A. Trusina, and M. Rosvall, "Hide and seek on complex networks," *Europhysics Letters*, vol. 69, pp. 853–859, 2004.

[17] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[18] M. Mamei, A. Roli, and F. Zambonelli, "Emergence and control of macro-spatial structures in perturbed cellular automata, and implications for pervasive computing systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, pp. 337–348, 2005.

[19] M. Jelasity and O. Babaoglu, "T-man: Gossip-based overlay topology management," in *Third International Workshop on Engineering Self-Organising Applications (ESOA'05)*, (Utrecht, The Netherlands), pp. 1–15, 2005.

[20] F. R. K. Chung, *Spectral Graph Theory*. Conference Board of the Mathematical Sciences CBMS, 92: American Mathematical Society, 1997.

[21] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat, "Lessons from three views of the internet topology," Aug. 2005.

[22] B. Bollobas, *Modern Graph Theory*. Springer, 1st ed. 1998. corr. 2nd printing ed., Aug. 2002.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.

[24] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs*. New York, NY, USA: Academic Press, Inc, 1979.

[25] H. Abu-Amara, B. A. Coan, S. Dolev, A. Kanevsky, and J. L. Welch, "Self-stabilizing topology maintenance protocols for high-speed networks," *IEEE/ACM Trans. Netw.*, vol. 4, pp. 902–912, 1996.

[26] T. S. E. Ng and H. Zhang, "Towards global network positioning," in *ACM SIGCOMM*, pp. 25–29, ACM Press, 2001.

[27] M. Costa, M. Castro, A. Rowstron, and P. Key, "Pic: Practical internet coordinates for distance estimation," in *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pp. 178–187, IEEE Computer Society, 2004.

[28] D. R. Karger and M. Ruhl, "Finding nearest neighbors in growth-restricted metrics," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, (Montreal, Quebec, Canada), pp. 741–750, ACM Press, 2002.

[29] M. Castro, P. Druschel, Y. Hu, and A. Rowstron, "Topology-aware routing in structured peer-to-peer overlay networks," *Lecture Notes in Computer Science*, vol. 2584, pp. 103–107, 2003.

[30] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Exploiting network proximity in distributed hash tables," in *Proceedings of the International Workshop on Future Directions in Distributed Computing (FuDiCo 2002)*, pp. 1–4, 2002.

[31] B. Zhao, A. Joseph, and J. Kubiatowicz, "Locality aware mechanisms for large-scale networks," in *In Proceedings of Workshop on Future Directions in Distributed Computing*, pp. 1–4, 2002.

[32] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 03, pp. 272–285, 2004.

[33] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *MobiCom*, pp. 56–67, 2000.

[34] P. Erdos and A. Renyi, "On random graphs," *Publ. Math. Debrecen 6*, vol. 6, pp. 290–297, 1959.

[35] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation.INFOCOM '96.*, vol. 2, pp. 594–602, 1996.

[36] K. Calvert, M. Doar, and E. Zegura, "Modeling internet topology," *Communications Magazine, IEEE*, vol. 35, pp. 160 – 163, 1997.

[37] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, p. 026118, 2001.

[38] D. S. Callaway, J. E. Hopcroft, J. M. Kleinberg, M. E. J. Newman, and S. H. Strogatz, "Are randomly grown graphs really random?," *Pysical Review E, cond-mat/0104546*, vol. 64, Apr. 2001.

[39] J. Kleinberg, "The small-world phenomenon: an algorithm perspective," in *STOC*, pp. 163–170, ACM Press, 2000.

[40] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," *cond-mat/0312028*, 2003.

[41] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–2, June 1998.

[42] H. A. Simon, "On a class of skew distribution functions," *Biometrika*, vol. 42, pp. 425–440, Dec. 1955.

[43] D. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, pp. 510–515, July 1965.

[44] D. Alderson, J. Doyle, R. Govindan, and W. Willinger, "Toward an optimization-driven framework for designing and generating realistic internet topologies," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 41–46, 2003.

[45] H. S. Wilf, *Generatingfunctionology*. A K Peters, Ltd., 3 ed., 2006.

[46] B. F. Cooper and H. Garcia-Molina, "Ad hoc, self-supervising peer-to-peer search networks," *ACM Trans. Inf. Syst.*, vol. 23, pp. 169–200, 2005.

[47] G. D. M. Serugendo, N. Foukia, S. Hassas, A. Karageogos, S. K. Most 'efaoui, O. F. Rana, M. Ulieru, P. Valckenaers, and C. V. Aart, "Self-organization: Paradigms and applications," in *Enginneering Self-Organizing Systems : Nature-Inspired Approaches to Software Engineering (ESOA 03)*, 2003.

[48] N. Forbes, *Imitation of Life: How Biology is Inspiring Computing*. MIT Press, 2004.

[49] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. K. Jr, R. Nagpal, E. Rauch, G. J. Sussman, and Ron, "Amorphous computing," *Communications of the ACM*, vol. 43, pp. 74–82, 2000.

[50] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kauffmann, 2001.

[51] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence From Natural to Artifical Systems*. Oxford Press, 1999.

[52] S. Wolfram, *A New Kind of Science*. Wolfram Media, 2002.

[53] D. N. Coore, *Botanical computing: a developmental approach to generating interconnect topologies on an amorphous computer*. PhD thesis, MIT, 1999.

[54] R. Nagpal, "Programmable self-assembly using biologically-inspired multiagent control," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, (Bologna, Italy), pp. 418–425, ACM Press, 2002.

[55] G. Csardi, "igraph: Routines for simple graphs, network analysis."

[56] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007.

[57] A. David, L. Lun, W. Walter, and C. D. John, "Understanding internet topology: principles, models, and validation," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 1205–1218, 2005.

[58] R. V. Sole and S. Valverde, *Information Theory of Complex Networks: On Evolution and Architectural Constraints*, pp. 169–180. Lecture Notes in Physics, Springer, 2004.

[59] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[60] M. Barthelemy, "Betweenness centrality in large complex networks," *EUR.PHYS.JOUR.B*, vol. 38, p. 163, 2004.

[61] M. Bauer and D. Bernard, "Maximal entropy random networks with given degree distribution," *cond-mat/0206150*, 2002.

[62] M. Kelaskar, V. Matossian, P. Mehra, D. Paul, and M. Parashar, "A study of discovery mechanisms for peer-to-peer applications," in *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing*, (Berlin, Germany), pp. 444–446, IEEE Computer Society, 2002.

[63] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Science, physics/0602124*, vol. 103, pp. 8577–8582, 2006.

[64] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E, physics/0605087*, vol. 74, p. 22, 2006.

[65] P. Pons and M. Latapy, "Computing communities in large networks using random walks (long version)," in *ISCIS2005*, Lecture notes in Computer Science, pp. 284–293, Springer-Verlag, 2005.

[66] A. Bellissimo, P. Shenoy, and B. N. Levine, "Exploring the use of bittorrent as the basis for a large trace repository," June 2004.

[67] "Gnutella peer-to-peer file search program." url: http://www.gnutella.org, 2007.

[68] Q. Lv, S. Ratnasamy, and S. Shenker, *Can Heterogeneity Make Gnutella Scalable?*, pp. 94–103. London, UK: Springer-Verlag, 2002.

Vita

Vincent Matossian

Education

2007 Ph.D. in Computer Engineering, Rutgers The State University of New Jersey, NJ, USA.

2003 MS in Computer Engineering, Rutgers The State University of New Jersey, NJ, USA.

1999 MS in Applied Physics, Université Pierre et Marie-Cury, Paris VI, Paris, France.

1998 BS in Applied Physics, Université Pierre et Marie-Cury, Paris VI, Paris, France.

Experience

2000-2007 Graduate Research Assistant, Center for Advance Information Processing, Rutgers University, NJ, USA.

1999 Web Developper, ClaraNet, Paris, France.

1999 Project Leader, Virtual Reality Laboratory, Rutgers University, New Jersey, USA.

1998 Algorithm Designer, Radiology and Image Processing Laborartory, Hospital Laennec, Paris, France.

Selected Publications

Towards Autonomic Control of Network Topologies. V. Matossian and Manish Parashar; Proceedings of Modeling Autonomic Communications Environments (MACE) 2006 workshop, Dublin, Ireland, October 25-26, 2006

Grid Computing in the Digital Oil Field. V. Matossian and M. Parashar; Hart's E&P August 2006, Vol. 79, No. 8, p 19-21

Content-based Decoupled Interactions in Pervasive Grid Environments. N. Jiang, C. Schmidt, V. Matossian, and M. Parashar, In Proceedings of BaseNets'04, San Jose, California, October 2004

Enabling Peer-to-Peer Interactions for Scientific Applications on the Grid. V. Matossian and M. Parashar; EuroPar 2003, Klagenfurt, Austria, August 2003

AutoMate: Enabling Autonomic Applications on the Grid. M. Agarwal, V. Bhat, Z. Li, H. Liu, B. Khargharia, V. Matossian, V. Putty, C. Schmidt, G. Zhang, S. Hariri and M. Parashar, Proceedings of the Autonomic Computing Workshop, 5th Annual International Active Middleware Services Workshop (AMS2003), Seattle, WA, USA, IEEE Computer Society Press, pp 48-57, June 2003

A Study of Discovery algorithms for peer-to-peer applications. M. Kelaskar, V. Matossian, P. Mehra, D. Paul and M. Parashar; CCGrid2002, May 2002

Engineering a Distributed Computational Collaboratory. S. Kaur, V. Mann, V. Matossian, R. Muralidhar, M. Parashar, Proceeding of the 34th Hawaii Conference on System Sciences, Hawaii, USA, IEEE Computer Society Press, January 2001

DISCOVER: An Environment for Web-based Interaction and Steering of High-Performance Scientific Applications. V. Mann, V. Matossian, R. Muralidhar and M. Parashar. *Concurrency and Computation: Practice and Experience*, John Wiley and Sons 2001;**13**8-9:737-754.