AN EVOLUTIONARY ANALYSIS OF THE INTERNET AUTONOMOUS SYSTEM NETWORK

A thesis submitted

to Kent State University in partial
fulfillment of the requirements for the
degree of Master of Science

by

Craig Robert Stewart

August 2010
Thesis written by

Craig Robert Stewart

B.S., Kent State University, 2008

M.S., Kent State University, 2010

Approved by

_________________________, Advisor
Javed Khan

_________________________, Chair, Department of Computer Science
Robert Walker

_________________________, Dean, College of Arts and Sciences
Timothy Moerland
Table of Contents

List of Figures ....................................................................................................................................... v
List of Tables ......................................................................................................................................... vii
List of Equations ................................................................................................................................... viii
Acknowledgments ................................................................................................................................. ix

1 Introduction ......................................................................................................................................... 1

2 Related Work ....................................................................................................................................... 7
   2.1 General Topological Analysis ........................................................................................................ 7
   2.2 Autonomous System Topology Analysis ....................................................................................... 8
   2.3 Network Analysis with Generating Functions .............................................................................. 10

3 Special Pattern Graphs ........................................................................................................................ 12
   3.1 Formula of the Fully Connected Mesh Ensemble ...................................................................... 13
   3.2 Formula of the Long Star Ensemble ............................................................................................. 13
   3.3 Formula of the Chain Ensemble ................................................................................................... 15
   3.4 Formula of the Trellis Ensemble .................................................................................................. 16

4 Generating Function Comparison and Similarity ............................................................................. 18
   4.1 Chain Similarity ............................................................................................................................ 21
   4.2 Star Similarity ............................................................................................................................... 22
   4.3 Trellis Similarity ........................................................................................................................... 23
   4.4 Homogeneous Forest of Congruent Patterns ............................................................................ 24
4.5 Homogeneous Forest of Non-congruent Patterns...............................................26
4.6 Heterogeneous Forest.........................................................................................30

5 Analysis and Results..............................................................................................33

5.1 General Analysis.................................................................................................36
  5.1.1. Scale of Growth.............................................................................................36
  5.1.2. Degree Evolution..........................................................................................36
  5.1.3. Peer Node Distribution.................................................................................39
  5.1.4. Clustering....................................................................................................41

5.2 Degree Distribution and Power-law Validity.......................................................43

5.3 Pattern Analysis..................................................................................................47
  5.3.1. Star Similarity Analysis...............................................................................49
  5.3.2. Trellis Similarity Analysis..........................................................................51

6 Conclusions.............................................................................................................55

6.1 Future Work.........................................................................................................57

References..................................................................................................................59

Appendix....................................................................................................................62
List of Figures

1.1 Topology Map of the Autonomous System Network in 2004.................................3

3.1 Illustration of Fully Connected Mesh......................................................................14

3.2 Illustration of Long Star..........................................................................................14

3.3 Illustration of Chain..................................................................................................16

3.4 Illustration of Grid Mesh..........................................................................................16

4.1 Example of Congruency and Similarity.......................................................................27

5.1 Example Link File.......................................................................................................34

5.2 Node Count (2004-09)............................................................................................37

5.3 Link Count (2004-09).............................................................................................37

5.4 Degree Distribution (2004-09)................................................................................38

5.5 Degree Distribution, pk = [0, 0.01] (2004-09)..........................................................38

5.6 Neighboring Node Distribution (2004-09).................................................................40

5.7 Clustering Coefficient (2004-09)............................................................................42

5.8 Clustering Coefficient (September 2008-09)............................................................42

5.9 Degree Distribution by Year (March 2004-09).........................................................43

5.10 Degree Distribution (2004-06)...............................................................................44

5.11 Degree Distribution (2007-09)...............................................................................45

5.12 Degree 2 Node Separation.......................................................................................48

5.13 Comparison of Significant Star Components..........................................................50
5.14 Star Similarity (2004-09).................................................................................................50
5.15 Comparison of Dominant Trellis Components..................................................................53
5.16 Trellis Similarity (2004-09)..............................................................................................53
List of Tables

5.1 Network Division Between Nodes Higher and Lower Than Degree 10..........................46
5.2 Dominant Trellis Dimensions (2004-09).......................................................................52
5.3 Dominant Trellis Dimension Testing...........................................................................52
List of Equations

1 Ordinary Generating Function Form.................................................................12
2 Fully Connected Mesh Generating Function Formula......................................13
3 Long Star Generating Function Formula.........................................................14
4 Chain Generating Function Formula................................................................15
5 Trellis Generating Function Formula...............................................................16
6 Error Minimization for Pattern and Network..................................................20
7 Node Count for Chain Pattern........................................................................21
8 Ratio for Long Star Pattern............................................................................22
9 Node Count for Long Star Pattern...................................................................23
10 Ratio for Grid Pattern....................................................................................23
11 Dimensions for Grid Pattern..........................................................................23
12 Homogeneous Forest Generating Function Form...........................................24
13 Pattern Count for Homogeneous Forest.........................................................25
14 Component Segmentation Equation................................................................27
15 Heterogeneous Forest Generating Function Form.........................................30
16 Neighboring Node Distribution Formula.......................................................40
17 Global Clustering Coefficient........................................................................41
18 Generating Function Expanded Form............................................................47
Acknowledgments

I would like to thank my advisor, Dr. Javed I. Khan, for all his help writing this thesis. He offered constructive criticism, guidance, and confidence to me throughout the entire process. His special topics course on complex networks which provided me with not only the inspiration for the topic of this thesis but also the opportunity to begin study in the field of networking.

I would also like to thank my thesis committee for their efforts. None of this would have been possible without them as well.

The Department of Computer Science also deserves my thanks for providing me with funding to complete my degree. The graduate assistantship I received was a great aid to me both financially and professionally. In addition, I have to thank Dr. Arden Ruttan for his help in acquiring this funding.

Last but not least, I thank my family. They provided me with both financial and emotional support throughout my college career, particularly during the time I was writing my thesis.
Dedicated to the memory of

Craig K. Wilson (1937 - 2009)

Mary Kay Petillo (1939 - 2009)

both of whom passed unexpectedly during my

graduate studies at Kent State University
Chapter 1
Introduction

The Internet is composed of autonomous systems (AS). An autonomous system is a part of the Internet – which itself is a network – that is owned and operated under one administrative domain. Autonomous systems are the basis of the distributed management of the Internet. Autonomous systems are part of networks owned and operated by regional and national ISPs, large organizations, and customers.

Long haul Internet routing is performed on the basis of autonomous systems. A well-defined, efficient routing system is necessary at the core of the Internet to ensure messages reach their destinations quickly and completely. The routing for the entities contained within an AS must follow a consistent routing policy [19]. The ASes use the Border Gateway Protocol, BGP, for routing between the ASes or domains. BGP is responsible for handling and broadcasting IP addresses to their peers. Since the implementation of each individual AS is hidden from view to protect the interests of its administrator or to prevent it from being the focus of an attack, IP addresses are not always visible so a unique autonomous system number is used to identify ASes instead.

This network of ASes is simply called the Autonomous System Network (ASN). The ASN is connected since all other systems on the Internet can reach the ASN in a measurable number of hops, the Internet can be viewed as the ASN alone. This fact
makes the understanding of the ASN vital to knowing the performance, growth, and traffic of the Internet. Without the complete map of the ASN, the entirety of the Internet cannot be perceived.

The analysis of the structure of the ASN is becoming more important as the need arises for its design to be more efficient and robust. For example, topological studies yield a variety of useful metrics related to the resilience of the network to attack such as in the face of cyber warfare or terrorism. Some networks rely on ensuring the safe transfer of messages between systems. The size of the network or the number of connections utilized within it is important in terms of the financial cost of building and maintaining large scale networks; networks that are overly connected could be rearranged to keep the transmission paths short but still lower the cost of running the network. Oliveira et al [11] writes: “It [the Internet topology] provides an essential input to the understanding of limitations of existing routing protocols, the evaluations of new designs, as well as the projection of future needs; and it will help advance our understanding of the interplay between networking technology, the resulting topology, and the economic forces behind them”.

Many researchers have recently done topological studies on the ASN [8, 10, 12, 17, 18]. The study of the autonomous systems poses many challenges, making it difficult to examine and monitor. First of all, the size of the ASN is very large. The number of autonomous systems was over 7000 and the peer connections between them were 15000 about ten years ago. In addition, both numbers are increasing rapidly – at times reported to be exponentially [4, 8] – on an annual basis indicating the increased complexity of the
Second, the study of structure is extremely difficult at such scale. Figure 1.1 shows the community structure of the ASN in March 2004. The network is complex – it has few recognizable topological features; the highly-connected nodes are in the center and the lower degree nodes branch out into points similar to a star topology. However, such visual recognition of structure has further degraded over time. The ability of the ASes to move and hide within the network radically changes the structure of the network. The very concept of any concrete pattern degenerates making it hard, if not impossible, to apply any deterministic graph traversal technique.

Finally, the ASN exhibits a high level of dynamism. The network is in a constant state of evolution. Many previous studies therefore have focused on one snapshot of the topology. The findings that utilize a single snapshot may not clearly provide all important information about the network.

The recent interest in power-law networks has also inspired studies into ASN complex network analysis techniques. Most of these recent studies of the ASN focus on
one snapshot. Few studies delve into the issues of topological and structural evolution. Other researchers dismiss trends in degree distribution entirely.

The novelty of this thesis is a formal technique involving ordinary generating functions. Their use is suggested for studying the evolution and structural characteristics of complex networks, and they are then applied to the analysis of the Internet ASN’s structural evolution from March 2004 to March 2009. This study has certain innovative properties to it. Though generating functions have been known for quite some time in mathematics, they are applied in the study of real-world networks only to test analogs of such networks. In addition, they are used here in particular to study the structure of a large scale complex network.

Generating functions capture some of these topological properties for specific networks from a statistical point of view, making them useful for the study of complex networks. Some significant distributions, like Poisson and exponential distributions, have been studied in the past. All generating functions that follow these distributions have unique qualities to them. However, familiar topologies like rings and stars have exclusive properties as well, but they are not used in the discussions of generating functions like the previously mentioned distributions. Since all Poisson graphs share similar generating functions and features, all familiar topological structures should follow the same trends as well. Because of this overlooked interpretation, generating functions should enable us to make statistical statements about these familiar topological structures and allow us to reveal the presence or absence of various dominant patterns in a very large graph.

This thesis demonstrates a general process detailing the ways generating functions
can be used to trace the presence of various topological patterns in a large scale network by searching for the signature of four specific topological patterns. Formulas are conceived to characterize the generating functions for each of these four structures. The patterns and formulas are then used to find the resemblance of actual networks to these structures and provide a means for measuring them. The proposed process can handle very large network graphs, yet the process is independent of the structure of the graph as it can be compared to any known topological structure.

With the aid of several known topological metrics and the developed technique, this thesis encompasses one of the most comprehensive studies of the structural evolution for the ASN. The data set used was obtained from Oliveira et al [6] for the analysis. This extensive trace of the ASN is built from data available from various AS observation sites such as BGP route tables and updates collected from large networks, such as RouteViews, RIPE-RIS, Abilene, CERNET route servers at Internet Exchange, BGP View, Packet Clearing House, UCR, traceroute.org, Route Server Wiki, and Looking Glasses like NANOG and the Looking Glass Wiki.

The thesis is organized in the following way. Chapter 2 covers recent research related to the topics of this thesis. Chapters 3 and 4 then explain the process by which generating functions can be used for structural property analysis in a complex graph. Four specific topological patterns are discussed prior to the method of searching for signatures of such patterns in a large complex network such as the ASN, including networks that mix multiple patterns. Chapter 5 presents the analysis of the ASN as it evolved during the 2004-2009 interval through the use of multiple metrics and the generating function
technique. Finally, this thesis is closed with conclusions and future work in Chapter 6.
Chapter 2

Related Work

2.1 General Topological Analysis

Many different methods currently exist for the evaluation of the topological structure of complex networks. Lee and Kalb [13] outline multiple such metrics in their paper. They measure the scalability, connectivity, and complexity of different network configurations using common metrics like average path length and diameter. These techniques were applied to several different topologies, ranging from familiar patterns like stars and rings to complex three- and four-dimensional structures like cubes and toroids.

Power-law regression is another common topic in network topological studies. Xie et al [16] completed a dynamic study of the power-law properties of the Gnutella P2P network. Based on the rank and degree distribution of the nodes, their study showed similarities to the use of generating functions. They used an ACC, absolute correlation coefficient, of 0.9 to determine whether or not the distributions complied with power-law. The higher the ACC was for the network, the more closely it followed power-law properties.

Some methods are created from the need to study specific types of networks.
Wang et al [14] used different metrics in their study of a DBE, a digital business ecosystem composed of a business network layer and agent peer-to-peer layer. They took numbers for the node count of the largest cluster in the network and its network availability, the percent of nodes reachable in five hops or fewer from a particular node. Chatterjee et al [15] presented an algorithm called the Weighted Clustering Algorithm (WCA) that came about from the study of ad hoc networks. Their algorithm revolves around finding nodes responsible for forming clusters, known as clusterheads, and choosing a number of those nodes that optimizes the degree of the network.

2.2 Autonomous System Topology Analysis

Several works can be traced which attempted to analyze the ASN. One of the earliest studies in this field was performed by Magoni and Pansiot [4], which analyzed six instances of the autonomous system network from November 1997 to May 2000 based solely on BGP data [4]. Magoni and Pansiot divided the network into a large mesh and surrounding trees that branch off the highly connected center. They found the node count in the network to reach a maximum size of over 7500, growing by 45% every year. The network had over 15000 edges at its highest, and that number increased by 53% on average each year. The average distance between nodes and the diameter remained constant throughout their study. They also declared that the number of distinct shortest paths and size of the trees followed power-law properties.

In 2002 Vazquez et al [8] used the autonomous system map from May 2001. This map was based off of BGP peer connections between the nodes, similar to the topological data obtained by Magoni and Pansiot. The node count reached over 11000 in that month,
a 46% increase on the final count reported in [4] one year earlier. The number of edges was marked above 23000, which was a 53% increase as well. In addition, Vazquez et al took measures of the clustering coefficient and average betweenness of the nodes; the clustering coefficient (taken as the ratio between number of edges and maximum possible value for a subgraph) was 0.22. Yook et al [10] recognized that the number of autonomous systems in their data set was over 12000 in mid-2002, only slightly higher than the data set of Vazquez et al one year earlier.

Zhang et al [17] built a topology map for an instance of the ASN in October 2005. Their combined snapshot after gathering data from the routing tables of RouteViews, RIPE-RIS, route servers, and looking glasses contained over 18000 nodes and 50000 edges. With nodes and edges that “disappeared” in the previous three months, their network size reached 19000 nodes and 60000 edges. They also claim the October 2004 instance of the ASN follows power-law degree distribution.

It can be noted that these first generation analyses are limited on snapshots. On the other hand, Dhamdhere and Dovrolis [18] study the evolution of the ASN over a span of ten years from 1998 to 2007. They found the numbers of nodes and edges to follow a linear regression rather than the exponential portrayed in [4], [8], and [10].

Clegg et al [12] established the Framework for Evolutionary Topology Analysis (FETA) for the study of evolutionary topology. FETA provides a fast process comparing a model of a network to its real-world counterpart and taking measurements on its evolutionary qualities. This framework was used to study the ASN from January 2004 to August 2008. Their study revealed information such as degree 1 and degree 2 node count,
maximum degree, and clustering based on the size of the network. Clegg et al [12] reported declining degree 1 node counts and increasing degree 2 node counts as the number of connections rose. In addition, the clustering coefficient remained less than 0.06 regardless of network size.

2.3 Network Analysis with Generating Functions

Although ordinary generating functions have not been used in applied-networks, there has been some previous work to understand various properties of large graphs using them.

One of the original uses for generating functions in networking was based on holding degree distribution information for calculating various qualities of random graphs. These random network graphs have nodes with arbitrarily assigned degrees based on probability. Newman et al [5] introduced degree sequences in the form of generating functions for this purpose. The distribution of probabilities is assumed to be normalized, giving a ratio between the sets of nodes of varying degrees. This method of creating random graphs allows researchers to study the topology of a network similar to the actual network with a degree distribution similar to the generating function used. This theoretical random graph is static and free of outside influences such as social phenomena that normally alter the real-life network.

Arbitrary random graphs were also used for determining clustering through percolation theory, the behavior of clusters (particularly connectivity) within the network. Callaway et al [1] use generating functions in order to find analytical solutions to site percolation in random graphs that were, once again, based on degree distribution. Their
formulation differs slightly from that used in [5]. Callaway et al utilized functions that were based on degree distribution and occupation of nodes. They provide a process for testing the robustness of a network to the random failure of nodes and links by manipulating the generating functions of their random graphs.

Generating functions are also useful in terms of directed network graphs. The master equation approach studied by Dorogovtsev and Mendes in [2] involves generating functions (called Z-transforms in their paper) for the degree distributions of networks that show preferential attachment to solve linear difference equations. This approach is an efficient method for finding solutions to network evolution problems.

Dorogovtsev and Mendes also developed a master equation that represents the in-degree distribution of a network over multiple iterations as a Poisson generating function. It was based on the distribution of nodes, number of edges added to the network, and additional attractiveness of nodes to receive those new edges.
Chapter 3
Special Pattern Graphs

The method of topological analysis proposed in this thesis makes use of ordinary generating functions (OGF). A generating function is a polynomial function that represents some aspect of a network. They most commonly contain the degree distribution information for the nodes in a specific network. The ordinary form is used, similar to the form used by Newman et al in [5]:

\[ G_0(x) = \sum_{k=0}^{\infty} p_k x^k. \]  

(1)

Each \( k \) value represents the degree while the \( p_k \) coefficients hold the probability that one node out of all nodes in the network has degree \( k \). For example, the generating function \( G_0(x) = 0.5x^2 + 0.5x \) contains information for a network that has an even distribution of degree 1 and degree 2 nodes.

Generating functions are used in this thesis as a means of identification of various patterns in the network. The generating function of a network is compared to the generating function for special patterns to determine their likeness to each other.

A pattern graph is a regular topological structure. Common examples of such patterns include stars and meshes. Four specific pattern graphs are taken into account, each of them and their formulas are introduced in the Sections 3.1-4.
3.1 Formula of the Fully Connected Mesh Ensemble

A fully connected mesh is a structure in which all nodes have edges to all other nodes. No self-cycles are considered, so the degree of all nodes in this pattern is one less than the total number of nodes in the pattern. This pattern yields a one-to-one correspondence between its generating function and topology (one generating function will produce the same random graph every time.) Figure 3.1 shows an example of a fully connected pattern of nodes.

**Theorem 1:** A fully connected mesh with \( n \) nodes has generating function

\[
G_{\text{mesh}(n)}(x) = x^{n-1}.
\] 

**Proof 1:** Consider a structure of \( n \) nodes. By definition of a fully connected mesh, every node has a degree of \( n-1 \). Since all nodes hold this value, the probability that a node has degree \( (n-1) \) is \( n/n \) or 1. Thus the generating function for this structure is \( x^{n-1} \).

3.2 Formula of the Long Star Ensemble

A star, as it is commonly known, is a structure of \( n \) nodes similar to a tree with the central node being degree \( (n-1) \) and the other \( n-1 \) nodes being degree 1. For the purposes of this thesis, this definition is designated differently. This modified star structure will be referred to as a *long star*. The points of a long star, consisting of the nodes branching off from the center of the structure, can be made from lines of nodes connected to each other. The number of points is its dimension, represented by the variable \( d \). Any function for a long star can generate many permutations of random graphs.
A long star has nodes divided into three categories: the center, the apexes, and the annexes. The center is the node in the star with the highest degree from which the points branch off. The apex and annex nodes make up the points of the long star. Apex nodes are always at the end of a point, have degree 1, and every point has exactly one. Annex nodes extend the points past their traditional length; a point has $x \geq 0$ annex nodes, each of which are degree 2. Figure 3.2 features an illustration of a dimension 7 long star consisting of 14 nodes.

**Theorem 2:** A long star pattern with $n$ nodes and dimension $d$ has generating function

$$G_{s(n,d)}(x) = \frac{1}{n} x^d + \frac{n - d - 1}{n} x^2 + \frac{d}{n} x.$$  \hfill (3)

**Proof 2:** The dimension of the star signifies the number of points on the star. According to the definition, this variable is equivalent to the number of apex nodes (degree 1) in the star. If there are $n$ nodes in the structure, then the chances of choosing an apex node randomly from all the nodes is $\frac{d}{n}$. The probability that the node is the center, or has degree $d$, is $\frac{1}{n}$. This situation leaves the annex nodes (with degree 2) with a $1 - \frac{d}{n} - \frac{1}{n}$ or
probability. Based on the earlier statement, the form of this generating function would be $\frac{1}{n}x^k + \frac{n-d}{n}x^2 + \frac{d}{n}x$. In the traditional star definition, the degree of the center node (designated by $k$ in the function) is $n-1$. This number must be adjusted to make up for the annex nodes added to the long star from the original. The degree of the center node is obtained by taking the number of annex nodes and removing them from the degree of the traditional center: $n-1-(n-d-1) = d$. The generating function for the long star is then $\frac{1}{n}x^d + \frac{n-d-1}{n}x^2 + \frac{d}{n}x$.

### 3.3 Formula of the Chain Ensemble

A *chain* is a group of ordered nodes linked only to the nodes that come before and after each other in the sequence. The first node and last node, having no node before and after it respectively, only connect to the nearest node in line. They are the only two nodes that are degree 1; all nodes between the first and last nodes are degree 2. As with the fully connected mesh pattern, a one-to-one correlation exists between generating function and the pattern it produces. Figure 3.3 provides an example of a chain of length 4.

**Theorem 3:** A chain structure of $n$ nodes can be produced using the generating function

$$G_{c,n}(x) = \frac{n-2}{n}x^2 + \frac{2}{n}x.$$  \hspace{1cm} (4)

**Proof 3:** Consider a long star pattern with dimension 2. The two apex nodes become the first and last in the chain sequence. All other nodes (center and annex nodes) are degree 2 by definition of the long star pattern. Therefore, the chain and dimension 2 long star have the same form so they may be considered the same structure. The generating function for
the dimension 2 long star is $G_{\text{ls}(n,2)}(x) = \frac{1}{n}x^2 + \frac{n-2}{n}x + \frac{2}{n}x$. By combining like terms, this equation results in $\frac{n-1}{n}x^2 + \frac{2}{n}x$ which simplifies to $\frac{n-2}{n}x^2 + \frac{2}{n}x$.

3.4 Formula of the Trellis Ensemble

The *trellis mesh* (or simply just *trellis*) is a system of nodes arranged like a square matrix. The edges of the pattern form a mix of squares and crosses between the nodes. The degree for any node cannot exceed the number of connections expected from a pure grid in which all edges in the trellis form squares. This pattern is defined by a height and width $p \times q$, requiring that $p \geq 2$ and $q \geq 2$. All nodes in this pattern have a degree of 2, 3, or 4 based on their position. One generating function for a trellis provides for many combinations of squares and crosses formed by the edges between each set of nodes. Figure 3.4 shows an illustration of a 3x4 trellis.

**Theorem 4:** A trellis structure with dimensions $p \times q$ has generating function

$$G_{g(p,q)}(x) = \frac{(p-2)(q-2)}{pq}x^4 + \frac{2(p+q-4)}{pq}x^3 + \frac{4}{pq}x^2. \quad (5)$$

**Proof 4:** A trellis of $p$ height and $q$ width will have $pq$ nodes total. All nodes on the
corners of the trellis are degree 2 while the rest of the nodes on the edges are degree 3. The remaining nodes, those on the inner portion of the trellis, are degree 4.

A trellis will always have four corner nodes to it. By definition a pattern cannot be smaller than 2 x 2 to be considered a trellis. Therefore, the $x^2$ term will have a coefficient of $\frac{4}{pq}$. The inner portion of the trellis is also a trellis itself surrounded by a row of nodes on all four sides, two lengthwise and two width-wise. Removing these rows of nodes gives an inner trellis of size $(p-2) \times (q-2)$, so the coefficient for the $x^4$ term is $\frac{(p-2)(q-2)}{pq}$. The remainder of the nodes are thus degree 3. This number can be found by subtracting the number of degree 2 and degree 4 nodes from the entire trellis. The number of degree 3 nodes is then $pq - ((p-2)(q-2) + 4) = 2(p + q - 4)$. Altogether the generating function becomes $G_{g(p,q)}(x) = \frac{(p-2)(q-2)}{pq}x^4 + \frac{2(p+q-4)}{pq}x^3 + \frac{4}{pq}x^2$. 
Chapter 4
Generating Function Comparison and Similarity

The process proposed in this thesis is called GFCS, Generating Function Comparison and Similarity. GFCS consists of identifying patterns within a network and comparing the generating functions of the network and pattern to determine their similarity. The result is presented as a percentage, describing the likeness of the familiar pattern to the network as a whole. The appendix of this thesis contains the entire algorithm for GFCS.

The process begins by analyzing the shape of the graph, utilizing it to discover its likeness to certain patterns. Network and pattern pairs must look similar to each other; they must share a similar signature, a common topological feature, with one another such as points of a star or a region of high clustering like a trellis. The terms in their generating functions also need to be comparable. Every exponent in the pattern must also exist in the network for the process to work properly. Two or more patterns may be chosen for comparison. These steps are carried out in lines 1-7 of the GFCS algorithm.

Once the patterns have been decided upon for analysis, the network generating function (the function for the degree distribution of the network) and the generating function formula (the formulas such as (2)-(5) that characterize the generating functions for all patterns within an ensemble using parameters) need to be compared in order to
find the right dimensions of the pattern graph. Comparisons are always done using the
*dominant coefficients*, the highest or most significant values from the generating function
formula. Constant values can never be dominant.

Dominant coefficients for a generating function formula \( G_0(x) = \sum f(n, m) \) are
determined by the significance of their parameters. Dominant coefficients are defined as
scaling with node count (i.e. being \( O(n) \)), or having non-zero parameters for very large
networks in which the limit on node count moves toward infinity. Consider a generating
function formula \( G_0(x) = \frac{n-m-c}{n} x^k + \frac{m}{n} x^{k-1} + \frac{c}{n} x^{k-2} \) where \( n \) is the total node count, \( m \) is a
distinct quality of the pattern that scales with the node count, and \( c \) and \( k \) are constants. If
the network is very large, the function becomes \( \lim_{n \to \infty} G_0(x) \approx r_1 x^k + r_2 x^{k-1} \) where \( r_1 \) and \( r_2 \) are
fractions of the remaining terms. While the degree \((k-2)\) term disappears, the other
coefficients remain large as both are based on the high node count of the network.

With the dominant coefficients decided upon, the parameters for the generating
function formula need to be calculated. The parameters can be determined by setting the
dominant coefficients for the generating function formula equal to the coefficients for the
matching terms in the network generating function. Always consider finding the node
count last among all parameters. The ratios of dominant coefficients are used in situations
involving multiple dominant coefficients. The parameters from the dominant coefficients
are then substituted into the generating function formula to obtain the specific *pattern
generating function* (the function for the pattern graph closest to the network.) The error
between the network generating function and pattern generating function is then
minimized. The error value, \( E \), can be calculated as:

\[
E = \sum_{k \in f} |a_k - f_k \Delta|
\]  \hspace{1cm} (6)

where \( a_k \) is the coefficient for the degree \( n \) part of the network generating function, \( f_k \) is the coefficient for the degree \( k \) part of the pattern generating function, and \( \Delta \) is the value that must be calculated to produce the smallest possible value for \( E \). The \( L^1 \)-norm is used here since a median of the limited terms is sufficient and for ease of calculation. The inversion of \( \Delta \) produces a percentage that represents the resemblance between the network and its matching pattern. The comparison of dominant coefficients is done in lines 17-21 of the GFCS algorithm while error calculation is completed in lines 23-25.

To illustrate this process, consider a graph of six nodes and generating function \( G_0(x) = 0.833x^5 + 0.167x^4 \). The first step in the process is to decide which pattern best matches this network. The network is too well-connected to be like a chain pattern. All nodes in the network have at least four connections while the maximum degree for a chain is two. The same statement is true for the long star in which all but one node has a degree lower than three. The trellis pattern is highly connected compared to the previous two patterns and shares an \( x^4 \) term with the network; however, the network lacks the \( x^3 \) and \( x^2 \) terms needed for proper testing. The fully connected mesh makes the best candidate in this case. The network only has six nodes, so its corresponding mesh would be degree 5, one less than the node count. The highly connected nodes and the \( x^5 \) term in the network match this pattern best.

The fully-connected similarity can be found for this example network. Fully-connected similarity is the term used to describe the percentage that represents the
resemblance of the network to a fully connected mesh. The generating function formula for the fully connected structure is (2). Since the generating function formula for the fully connected mesh only has one term, only one variable is needed. In order to compare the network to a fully connected mesh, the total number of nodes is needed to determine the term that will be compared. The dominant coefficient in the generating function formula is the coefficient for $x^{n-1}$ since it has no other terms. Only the $0.833x^5$ term in the network generating function will be needed to find the fully-connected similarity of the network. The value 1.2 for $\Delta$ in the following solution will minimize the error for the network: $E = 0 = |1 - (0.833)(1.2)|$. The fully-connected similarity of this graph is thus 83.33% since $\Delta^{-1} = 1/\Delta = 0.8333$. No other calculations need to be made for fully-connected similarity.

Some parts of the process are changed to suit each pattern ensemble. The methods for finding chain, star, and trellis similarity will illustrate the differences.

4.1 Chain Similarity

Equation (4) is the generating function formula for chains. This formula only contains $x$ and $x^2$ terms and the $x$ term has a dominant coefficient. The only dominant value here is the $x^2$ coefficient, and it must be set equal to the $x^2$ coefficient in the network generating function. The first objective is to find the number of nodes, $n$, which can be achieved by solving the equation:

$$\frac{n-2}{n} = a_2 \text{ or } n = \frac{2}{1-a_2}.$$  

(7)

The value of $a_2$ is the $x^2$ coefficient in the network generating function. Replace $n$
in the generating function formula with the value calculated from (7) to obtain the pattern generating function. Finally minimize the error between the two sets of coefficients for the x and x^2 terms and invert the Δ-value to find the chain similarity of the network.

4.2 Star Similarity

The generating function used here is (3). This formula has three terms; however, the x^d coefficient is 1/n so it is deemed non-dominant. The x^2 term is dominant since it relies heavily on the node count. The x term is also considered dominant. A long star pattern can be made larger by increasing the length, dimension, or both. Due to the fact that dimension contributes at least partially to the total number of nodes in the pattern, this parameter can be considered O(n). The x term relies on the dimension, so it is considered dominant by definition. The coefficient of the x^2 term is the largest, and the coefficient for the x term is then the second largest. As stated earlier, the ratio of the most dominant and the second most dominant coefficients are taken when two dominant coefficients exist. They are then compared the same way as any other pattern graph:

\[
\frac{n-d-1}{d} \frac{a_2}{a_1}. \tag{8}
\]

A number of nodes and dimension that best fit in this formula are required. The potential value for dimension can be taken straight from the network generating function. The dimension value that should work best in this case is the x coefficient of the network generating function multiplied by the total number of nodes in the network (\(\frac{a}{d} * n\)). The node count can then be decided by solving for n in (8).
\[ n = \frac{a_2 d}{a_1} + d + 1 \]  \hspace{1cm} (9)

Any fractional part of the \( n \) value can be removed from the result of (9) to form the \((d,n)\)-pair. These values are then used to create the pattern generating function by substituting them into the generating function formula. The rest of the process is the same as outlined before: find the \( \Delta \)-value that produces minimal error between the \( x, x^2, \) and \( x^d \) values of the network and pattern generating functions and then invert that value.

4.3 Trellis Similarity

Based on (5), the \( x^3 \) and \( x^4 \) parts are the most dominant in the trellis structure as they are based on the dimensions of the trellis. The \( x^2 \) term, again being a constant value over the node count, does not carry anything significant. The ratio is then the \( x^4 \) to \( x^3 \) coefficients:

\[
\frac{(p-2)(q-2)}{2(p+q-4)} = \frac{a_4}{a_3} \hspace{1cm} (10)
\]

The \( a_4 \) and \( a_3 \) variables represent the \( x^4 \) and \( x^3 \) coefficients of the network generating function.

In order to obtain the dominant values, some error testing is necessary. A value for \( q \) within the range \( 2 \geq q \geq \) diameter of the network must be found that produces the lowest error from the nearest integer. In other words, the \((p,q)\)-pair that creates the best fit for (10) is needed. The following equation can be used for error testing:

\[
p = 2qr + 2q - 8r - 4 \hspace{0.5cm} \text{for} \hspace{0.5cm} r = \frac{a_4}{a_3}. \hspace{1cm} (11)
\]

It is advised to remain within a threshold of \( \pm 0.1 \) when determining proper values
for $p$. Since both $p$ and $q$ are equally significant in these functions and their values need to be integers, it is important to find the values that best fit the dimensions of the dominant trellis in the network. The error testing is the same as the other structures.

### 4.4 Homogeneous Forest of Congruent Patterns

A network can potentially be built from many *congruent* patterns, graphs that have the exact same generating function. These pattern graphs could be linked together by additional edges in some manner or disjoint from each other. This type of network would be considered a *homogeneous forest*. Homogenous forests are not pure pattern graphs, but the analysis of a pattern's likeness to such a forest can potentially reach 100% (the equivalent of the network being the exact same as the pattern graph.) Only forests made up of disjoint trees, the individual patterns that make up the forest, can achieve 100% pattern similarity. A homogenous forest is based on the generating function:

$$G_{f(n)\ast m}(x) = mf(n). \quad (12)$$

The value of $m$ is the number of patterns and $f(n)$ represents the generating function for one of the trees. Since generating functions hold the ratios of nodes of different degrees in the network, creating a network of like patterns will cause no change in the pattern generating function.

**Theorem 5:** The value of $m$, the number of congruent patterns in a homogeneous forest, does not make the generating function for the homogeneous forest different from that of its individual patterns.
**Proof 5:** Consider a pattern graph with generating function $G_0(x) = \sum_{i=0}^{\infty} b_i n x^i$ where all $b_i$ are the number of nodes of degree $i$ and $n$ is the node count for the pattern. A homogeneous forest of $m$ patterns will have $m n$ nodes. Each set of nodes of degree $i$ will also have $m b_i$ nodes in it. The homogeneous forest would then be $G_T(x) = \sum_{i=0}^{\infty} \frac{m b_i}{m n} n x^i$. The $m$ values cancel themselves out, leaving $G_{T0} = G_0$.

In order to separate similar patterns out of a graph, it is important to determine the number of patterns first. Some homogeneous forests will clearly show this count in a complete view of the graph; still, it can always be determined mathematically through the pattern generating function. The constant values, contained within the non-dominant coefficients, are needed in this situation. The constants for a pattern can be obtained by multiplying the non-dominant coefficients from the generating function formula by the total number of nodes in the pattern, $n_f$. The same can be done for the non-dominant coefficients of the network generating function and the node count for the network, $n_a$, to find the same constant multiplied by $m$. When the constant portion of the network generating function is divided by the corresponding part of the generating function formula, the $m$ value can be observed, represented as:

$$m = \frac{a_c n_a}{f_c n_f}$$  \hspace{1cm} (13)
each coefficient in the network generating function by \( m \), the node count within each degree set of a single pattern can be calculated.

When determining the likeness of a homogeneous forest to a pattern, there is no difference from searching for a pure pattern as in Sections 4.1-3. The process relies on comparing generating functions. Because the ratio of the network generating function is no different from the pattern to be searched for, no additional steps are involved in the process. However, some patterns do rely on the node count for each term of the pattern. For example, finding the star similarity of a network graph depends on the dimension of the star, obtained by multiplying the \( x \) coefficient by the node count.

4.5 Homogeneous Forest of Non-congruent Patterns

Graphs consisting solely of congruent patterns are rare and often times not useful in analysis. It is much more common for a network to be built from a mesh of multiple, similar but non-congruent patterns. The term similar is used here to describe two pattern graphs that come from the same ensemble. Congruent patterns are similar by definition. The generating functions for two non-congruent patterns will differ in at least one of the coefficients or parameters. It is necessary in this case to isolate each individual pattern before continuing with analysis.

In order to isolate patterns from the network, the parameters of each pattern and their individual contributions to the network must be found. In other terms, for the following equation representing a coefficient, \( C \), in a network of similar but non-congruent patterns:
$$C = \sum_{i=1}^{m} s_i G_p(x)$$

(14)

the values of the parameters for each $G_p(x)$, the coefficient from the generating function for each similar pattern, and $s_i$, the fractional contribution of the pattern to the network, must be solved for $m$ patterns.

This process begins by determining the number of patterns in the network. Due to the fact that the patterns are different from each other, the method for finding the $m$ value in Section 4.4 will not produce an exact count. The $m$ value can instead be used as an approximation on the number of patterns. For functions like the long star that rely on the node count, a maximum value on the total nodes of the pattern can be set instead. This effectively creates a cap on the number of patterns found in the forest.

The next step is expanding the coefficients of the network generating function as in (14). Each $G_p(x)$ should be represented as they would in its generating function formula. The key here is finding the unique values for the parameters of each pattern in order to form the pattern generating functions. The new generating function created from

---

**Fig. 4.1:** An illustration of congruency and similarity. The two patterns on the left are congruent. They are from the same ensemble and have the same generating function. The two patterns on the right are non-congruent but similar. They are from the same ensemble but have different functions.
this process will be referred to as a segmented generating function. Equations should then be formed from this segmented generating function by setting the coefficients from the original network generating function equal to the new coefficients. If several coefficients in the network could be set to the expanded coefficient, start with the largest of those values and work down. The ratio of dominant coefficients and all other equations in the system should be used to determine whether or not the term is part of a pattern (usually with a threshold of ±0.1 similar to trellis similarity analysis in Section 4.3.) Next, solve for all variables using this system of equations. The parameters make up the set of variables that must be solved. Once again, the parameters are used to create the pattern generating functions for each pattern. Finally, a pattern generating function is created for comparison to the network generating function. Forming the coefficients for this function can be achieved by taking the node counts of the coefficient for each pattern (done by multiplying each coefficient by the node count for that pattern), summing together the numbers for each coefficient, and dividing by the combined total of all nodes in all patterns. With the pattern and network generating functions, the rest of the process involves minimizing the error and inverting the common multiplier as with all previous analysis methods.

The difficulty by which these equations can be solved lend to three different solvability categories for patterns: inseparable, separable, and separable with difficulty. Inseparable graphs are patterns in which the parameters do not give enough information to create a reliable isolation of the individual pattern graphs in a network. The most common set of inseparable graphs are those that rely on node count as their only
parameter as the number of nodes in each pattern is usually unknown and unable to be calculated without further information. Separable graphs are patterns in which the number of parameters or usefulness of information in the parameters makes it easy to isolate individual patterns from a network. At least one of the the parameters in the generating function formula must show some direct correlation to the node count or other parameters in the function. Graphs that are separable with difficulty are patterns with parameters that are too few in number and lack the information to be useful. In theory, these patterns are separable; however, they rely on information relevant to the specific network containing them. As a general rule, one equation needs to be formed for every parameter used in order to solve for each parameter. If the number of patterns is too large or the number of parameters for the pattern is high, then it is possible that not enough equations could be formed to meet the quantity of data needed. Theoretically the parameters can be found if more information about the network, such as diameter or average path length, is known. Nonetheless the parameters must be able to contribute to these metrics. Other useful information, like the ratio of dominant coefficients, could be useful in this calculation.

Three of the special patterns discussed in Chapter 3 – the chain, the long star, and the trellis – easily fit into each of these solvability categories. The chain is a prime example of an inseparable pattern. It relies only on node count, and the only information held in the coefficients of a pattern generating function for a chain is the number of degree 1 nodes, a constant. With only a constant and an unknown variable, there is no way to solve equations dealing with chains. The long star is separable. The dimension is
the key parameter here as it lends itself not only to the node count but also one of the exponents. Each individual long star can be pulled from the network by taking its exponents and determining whether or not they form stars by determining their node counts using the ratio of annex nodes to apex nodes as in (9). The trellis pattern is separable with difficulty. To separate two trellis patterns from the network, one could derive the following set of equations from (14):

\[ C_4 = s_1((p_1 - 2)(q_1 - 2)) + s_2((p_2 - 2)(q_2 - 2)) \]
\[ C_3 = s_1(2p_1 + 2q_1 - 8) + s_2(2p_2 + 2q_2 - 8) \]
\[ C_2 = 4s_1 + 4s_2 \]

The \( C_n \) are the coefficients of the \( n \)th term. Six parameters must be solved for here \((s_1, p_1, q_1, s_2, p_2, q_2)\) but none of them hold any information useful to the other variables. In theory, the ratio could also make a sufficient equation to add to this system. The diameter of the network might be useful as well depending on the configuration of the network. The values of \( p \) and \( q \) must be less than the diameter of the network, so it could also add insight to the individual parameters of each trellis.

### 4.6 Heterogeneous Forest

Some networks can have qualities like two or more dissimilar patterns. The term dissimilar refers to graphs that have different structures or generating functions with different terms. The function for such a network usually comes in the form:

\[ G_{\{h(n)\} \in F}(x) = \sum h(n) \].

This formula represents a network with a set of patterns, \( F \), each with its own generating function \( h(n) \).
In the case that multiple patterns are detected in the same network graph, these patterns must be separated from each other to make comparisons between the network and patterns. This separation only needs to occur when any two of the patterns share a common term. A set of patterns that do not share any common terms are already separated; however, a network that shows signatures of two or more patterns that have at least one common exponent in their generating functions must divide their coefficients between the two pattern graphs. In short, no two pattern graphs can share the same nodes. Usually the patterns and network are only similar in their generating functions. A mathematical approach to divide the nodes of the network between the patterns is necessary. The division of nodes is represented by lines 12-16 in the GFCS algorithm.

This approach to separating the network between two patterns starts by determining the first pattern to isolate. Separable structures should be isolated before patterns that are separable with difficulty. If more than one pattern is separable or separable with difficulty then patterns with a one-to-one correspondence between function and random graph should be considered last. Once a pattern is chosen, separate all subgraphs of that pattern from the network using the method in Section 4.5. The patterns should continue to be tested until the sum of their node counts reaches a maximum, based on some quality of the generating function formula. When one set of patterns is isolated from the network, choose another one and repeat the above steps. The process of separating patterns from the network continues until only one pattern remains. All remaining nodes for the shared terms contribute to that term in the final pattern. The parts of each shared coefficient are split between the patterns that share them to create
their pattern generating functions. The sum of all generating functions for the same pattern class is used to determine the likeness to a pattern in the same manner as homogenous forests of non-congruent patterns.

This process requires that the set of pattern ensembles have at least one unique term among all the generating functions. If a term is shared between any two generating functions then it is not considered unique. An unshared term is needed to determine which nodes of a shared term in one pattern graph belong to that pattern.
Chapter 5
Analysis and Results

The previously outlined analysis method has been applied to analyze the Autonomous Systems Network. The following sections outline the experiment and provide results for many different metrics. The initial analysis from Sections 5.1 and 5.2 detail the degree distribution of the autonomous systems to provide a clearer view of the network along with additional information for the generating function analysis of which the results are presented in Section 5.3.

Data from [6] was used to compose the data set for several iterations of the ASN for the analysis. This is the most extensive trace of the ASN available to-date. This rich data set combines (a) BGP advertisements and updates observed at various AS observation sites such as large networks including Route Views, RIPE-RIS, Abilene, CERNET, (b) route server data at various Internet Exchanges such as BGP View, Packet Clearing House, UCR, traceroute.org, Route Server Wiki, and (c) data at Looking Glasses such as traceroute.org, NANOG, Looking Glass Wiki.

The data set did not combine any trace route data since no accurate method of interpreting AS paths from IP paths currently exists. A topology map is built each day from these BGP routing tables, route servers, and Looking Glasses. All data is obtained with publicly available software that allows for the viewing of routing information for the
autonomous system of a single entity. The data, however, is not perfect. Oliveira et al [6] note that this is a conventional technique that can potentially miss links and nodes due to lack of information in routing tables and the unwillingness of service providers to allow their nodes to be discovered, most of which included peer-to-peer links [7].

The data set taken from [6] contains the nodes and links from the first day of every third month for all viable years, ending with the month this project began. The collected maps span over every March, June, September, and December from March 2004 (the first year the maps were created) to March 2009 (the date this project began.) Although more data is available, these 21 maps are used in the data set for this project.

Part of one of the files used in this data set is displayed in Figure 5.1. Each iteration of the network is divided into two files. One carries the list of monitored nodes while the other contains a list of links. The records for these files start with the unique identification numbers of the ASes assigned by the IANA. The node files have one ID and the link files have two IDs, one for each AS connected to that link. The next two fields are the times the node or link was first and last monitored by the software, represented as Unix timestamps (seconds since the epoch.) The last piece of information represents the node or link's position in a customer-provider chain. A 0 is used for the

Fig. 5.1: An excerpt from the link file of the March 2004 ASN map.
In order to analyze the collected data set, a program was developed to obtain more information about the network at each month in the study. These programs were written in Java using the JUNG (Java Universal Network/Graph) Framework, a software library for the modeling, visualization, and analysis of network graph data. JUNG was used not only to build a representation of the network inside the computer but also to access its powerful measurement tools. The program is divided into two parts: the graph builder and graph analyzer.

The graph builder was written for the purpose of reading in the node and link files for each individual month of the study and creating the network graph from that data. It builds an undirected graph by adding each node and link to the graph object. The program also checks for duplicate nodes and links, including reverse paths, before adding them to the graph object. In addition to the network graph, the graph builder keeps a link matrix for the graph. The link matrix acts as an adjacency matrix, a look up table for determining the existence of a link between two nodes in a time and space efficient manner.

Once a graph object is created, it is passed from the builder to the graph analyzer for measurement. In addition to obtaining the degree distribution for each iteration of the network, the graph analyzer also has the capability to measure other aspects of the graph object such as clustering. It can also produce a visualization of the graph using a “self-organizing map layout algorithm, based on Meyer's self-organizing graph methods” [3].

This thesis presents the analysis of the dynamic evolution characteristics of the
ASN. First, known metrics depict general characteristics of the network. A detailed analysis of the degree distribution evolution is then provided. The analysis of topological evolution is presented last.

5.1 General Analysis

5.1.1 Scale of Growth

Figure 5.2 plots the growth in terms of number of ASes traced in the Internet. The number of autonomous systems steadily increased over each three-month period adding about 5000 ASes each year. The node count more than doubled over the course of five years. Figure 5.3 plots the growth of links representing paths between peers inside the ASN. The number of links also increased linearly but almost fivefold in the same period. Both growths seem to be following a linear trend with Pearson correlation $R^2 > 0.98$ throughout the period of study. This is markedly different from previous studies which speculated exponential increases in both quantities by 45% and 53% each year respectively [4, 8]. The growth of the ASN is on par with the Internet itself. Due to the fact that it is the core routing system, the ASN must continue to meet the growing demand placed on the Internet. The number of ASes must increase to handle the demand, and the number of peer connections must similarly rise to maintain or improve upon past routing speed.

5.1.2 Degree Evolution

The degree distributions showed some interesting results, illustrated in Figure 5.4. Each block represents the $p_k$ value, distribution for nodes of degree $k$, for the
corresponding degree and month in the study as indicated by the axes. White blocks represent $p_k$ values of 0. The more nodes in a degree set, the more the block becomes black (representing $p_k = 0.5$ in Figure 5.4 and $p_k = 0.01$ in 5.5.)

The degree 1 and degree 2 node counts were the highest. They show up in Figure 5.4 as distinct, dark colored bands that grow lighter over time. This trend shows that the
ASN is slowly moving away from the stub configuration, in which an autonomous system remains based on one or two connections. The bands for the degrees between 3 and 10 follow the opposite pattern, starting light and gradually becoming darker as time progresses.

**Fig. 5.4:** The degree distribution for each iteration of the network. Darker grays and blacks are closer to $p_k = 0.5$. White blocks are $p_k = 0$.

**Fig. 5.5:** Same as Figure 5.4. Darker grays and blacks are closer to $p_k = 0.01$.
moves on. These changes indicate that autonomous systems with one or two peers appear to be obtaining more links to providers or peers over time. In order to show the trends at degrees higher than 10, the scale was reduced as in Figure 5.5. Going across the chart from month to month, the contrast of the bands varies at all values of $k$. It can be seen that the darker colored blocks (the distributions of higher $p_k$) are moving higher over time, causing diagonal bands as opposed to the horizontal bands seen in the first chart. This observation is supported by the fact that all values greater than 0.01, the black blocks at the bottom of the chart, seem to rise in a similar diagonal manner as time moves on.

Most of the trends suggest a steady evolution of the network; the increases in the numbers of autonomous systems, customer-provider connections, and peer connections are steady and predictable. Nonetheless, the ASN makes a surprising change between months 57 and 60. During this interval at the end of 2008, all blocks below degree 2 become significantly darker and all blocks above degree 3 become noticeably lighter than with all previous intervals. The maximum degree jumped from 7000 to over 10000 as well. A higher density of high degree nodes also exists. The ASN appears to have experienced a period of accelerated growth during this time period, making it increasingly difficult to monitor the network as a whole.

### 5.1.3 Peer Node Distribution

Generating functions have the ability to reveal distributions of other complex properties about a network through mathematical derivation as outlined in [5]. One such quality of interest is the number of nodes reachable in one hop. This measure of the ASN
40 is likened to the size of the neighborhood or AS clique. The higher the distribution of neighbors for an AS, the greater its connectivity and ability to route traffic to other parts of the network quickly. The neighboring node distribution is obtained by taking the derivative of the generating function for a network:

\[ G_1(x) = \frac{1}{z} G_0'(x) \] (16)

where \( z \) is the average degree of the network, which can be obtained by taking \( G_0'(1) \).

Figure 5.6 details the distribution of neighboring nodes for each year of the study. Each line represents one year, showing the number of neighbors and distribution of nodes with those counts for each degree. These numbers follow an almost parabolic trend. The degree 1 neighbor distribution decreases at a faster rate than the degree 2 node distribution from the previous subsection. The rest of the numbers are almost the same as the degree distribution patterns up to the point they become close to 0 before the 100

**Fig. 5.6:** The distribution of neighboring nodes in the network based on the generating functions for March of each year. The years are represented by the individual lines on the graph.
neighbor mark. Beyond nodes with 100 neighbors, the distribution increases exponentially.

5.1.4 Clustering

Clustering exposes the amount of organization of a large network. This metric shows two important aspects of the network. For one, networks with high clustering tend to resist random attack better, being able to provide services despite few ASes becoming unusable. Second, high degrees of clustering provide insight into the hierarchy of the network. For example, groups of clustered tier-1 autonomous systems, which act only as providers, tend to be common in the ASN. The global clustering coefficient is commonly used to measure the clustering of a network. This metric is represented as $C_M$ for month $M$ and measured as:

$$C_M = \frac{3 \times (\text{number of triangles})}{(\text{number of triples})}$$

in which triangles are sets of three fully connected nodes and triples are sets of three nodes connected by links in any manner.

Figure 5.7 shows the change in clustering coefficient over time. The clustering coefficient for each month studied followed a predictable pattern as it is increased constantly. The largest increase in clustering occurs between March 2004 and December 2004 in which it grows from about 0.037 to 0.056.

Similar to the degree distribution numbers, an interesting change occurred at the end of 2008. In the general analysis, the clustering decreased by 0.025 since the beginning of the year 2009. Further investigation was conducted into the clustering
coefficient for this period. The results of this investigation are shown in Figure 5.8. The total drop in clustering was roughly $\Delta C = 0.065$, taking place between October 2008 (month 58) and February 2009 (month 62) before reverting back to its original steadily increasing trend. These numbers support the earlier claim that the ASN went through a
period of rapid growth during this time; additionally, this expansion took place during a four-month span.

5.2 Degree Distribution and Power-law Validity

It has been claimed previously that the degree distribution of the ASN follows power-law. Consequently, this thesis focuses particular attention to this study. The degree distribution over the course of the study, however, suggests the network now deviates significantly from power-law. Figure 5.9 provides a perspective on the degree distribution that shows this trend. An obvious curve in the chart exists from degree 2 onwards even though the axes are scaled logarithmically. The numbers drop off heavily from the start, and the difference between one degree and the next decreases steadily over time.

Further evidence of the lack of power-law in the ASN is shown in Figures 5.10 and 5.11. The function $f(x)$ on each chart is the closest power-law line to the graph and $R^2$.
Degree Distribution (2004-06)

Fig. 5.10: The degree distributions of the ASN for 2004-2006.
Degree Distribution (2007-09)

Fig. 5.11: The degree distributions of the ASN for 2007-2009.
is the calculated Pearson correlation coefficient between the actual numbers and the power-law trend. Correlation numbers of 0.9 [9] and 0.95 [4] were used in past works as a baseline for determining the existence of power-law in a network graph. The Pearson correlation coefficient is used here as the absolute correlation coefficient. The graphs of the ASN for all years do not come near either of the suggested values.

Signatures of power-law can only be confirmed for a partial degree distribution, in which only a portion of the terms – the degrees used in the distribution – are considered. Magoni and Pansiot [4] reported on nodes between degrees 1-10 in their findings. They also stated that the degree 1 nodes should not affect the power-law properties of the network. Taking these two notes into consideration, if only the nodes with partial degrees between 2 and 10 are considered, a strong correlation to power-law (0.9976) is observed. However, the number of nodes of degrees higher than 10 are becoming a much more significant portion of the network. Table 5.1 illustrates the percentage of the ASN consisting of these high degree nodes. As evident, they are rapidly increasing – from

<table>
<thead>
<tr>
<th>Year</th>
<th>Nodes of Degrees 1-10</th>
<th>Nodes of Degree 10+</th>
<th>Percent of Nodes Degree 10+</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>16243</td>
<td>1116</td>
<td>6.43%</td>
</tr>
<tr>
<td>2005</td>
<td>19686</td>
<td>1842</td>
<td>8.56%</td>
</tr>
<tr>
<td>2006</td>
<td>23098</td>
<td>2547</td>
<td>9.93%</td>
</tr>
<tr>
<td>2007</td>
<td>26474</td>
<td>3366</td>
<td>11.28%</td>
</tr>
<tr>
<td>2008</td>
<td>29794</td>
<td>4131</td>
<td>12.18%</td>
</tr>
<tr>
<td>2009</td>
<td>33516</td>
<td>5249</td>
<td>13.54%</td>
</tr>
</tbody>
</table>

Table 5.1: The number of nodes with degrees lower and higher than 10 for March of each year in the study.
6.4% in year 2004 to 13.5% in year 2009 – which almost invalidates the analysis based on partial degree distributions.

Delving further into this idea of partial power-law, the correlation for the degree distribution of the the degrees in the range 1-272 in the ASN to power-law is about 0.9. This set barely makes a weak-power law. If nodes beyond degree 272 are included, the correlation falls off sharply. Thus it is clearly evident that the network is hardly a weak power-law when the concept of partial degrees is considered, but even then it is rapidly moving away from that trend.

Previous research on the topology of the ASN argued the existence of power-law. Zhang et al [17] insist on the power-law aspects of the Internet influencing the ASN. Magoni and Pansiot [4] acknowledge that power-law for the network only applies to out-degree on a directed graph. Many other properties of the ASN do follow power-law as proven in [4], but no one has been fully able to prove its existence for degree distribution. According to research presented in this thesis, this factor of the ASN is weak but improving.

5.3 Pattern Analysis

The degree information and node counts stored for the graph of each month in the study were used to build the generating functions for each one. The functions were written in the form

\[ G_0(x) = \sum_{k=1}^{\infty} \frac{t_k}{n} x^k \]  \hspace{1cm} (18)

where \( k \) is the degree, \( t_k \) is the total number of nodes that were of degree \( k \) in the network.
for that month, and \( n \) is the node count for the entire network. The generating functions were formulated this way to preserve each piece of data for future use.

The generating functions, metrics, and visual graphs were used to determine the pattern graphs to be tested. Among the many observations, the ASN shows signatures of the long star and the trellis mesh ensemble as the \( x, x^2, x^3, \) and \( x^4 \) terms of the generating functions were all fairly high. These two patterns share the \( x^2 \) term, however, so the degree 2 nodes would need to be divided between the two patterns.

Figure 5.12 shows the separation of the degree 2 nodes in the ASN for the long star ensemble. Since the long star is under the separable solvability category, the degree 2 nodes for the long stars were separated from the nodes that did not belong to them. The leftover nodes would be attributed to the trellis mesh ensemble. Due to the difficulty of separation for trellises, each iteration of the network was considered to contain only one trellis when determining their trellis similarity percentages. For the separation of the star
nodes, the system of equations is not used here since the $x^4$ term is most significant in determining the star parameters and too many large degree numbers exist for the number of degree 1 nodes. In this case, all degree values less than 5% of the node count were used as potential dimension numbers in order to limit $m$. The dimension values were substituted into (9) and then tested for error from the nearest integer. If the error was under 0.1 then the dimension and node count were kept as part of the pattern generating function.

The view of the network as a heterogeneous forest of long stars and one trellis weighs heavily on the separation of degree 2 nodes in the network. Figure 5.12 shows that the majority of these nodes are part of the long star structures in the network. The trellis pattern has a small constant for the $x^2$ term in its generating function, so the low numbers help the error testing phase of trellis similarity calculation.

5.3.1 Star Similarity Analysis

The visualization of the early network graphs (as in Figure 1.1) showed significant presence of long star like ensembles at the edges of the network. Figure 5.13 shows the degree measures for significant components of the long star ensemble i.e. the numbers of nodes with degree 2 and degree 1 within the ASN. These were higher than all other numbers and several centers appeared to exist as many of the high degrees had $p_k$ values of $1/n$. These two aspects made finding the star similarity of the ASN favorable.

The dimensions and sizes of the long stars within the ASN loosely followed few trends. The stars did have a tendency to grow over time as they obtained more points; however, the higher the number of nodes the star accumulated, the more connected it
The star ensemble analysis of the network exceeded the 50% mark for the first year of the study; however, the number ended up dropping below that mark for the remainder of the study. The star similarity for March 2004 was nearly 65%. The next two
months remained close to 60% before a sharp decrease occurs at the end of 2004. Month 15 was the first month in the study in which the star similarity value dropped below 50%. The star similarity percentage reaches its lowest point in March 2009 when it is slightly higher than 34%.

Figures 5.13 and 5.14 compare the dominant long star terms from the ASN against the calculated star similarity percentages for each iteration. Both charts show a steady decline in all values over the course of the study. Some differences are notable between the two statistics. The $x$ and $x^2$ terms of the ASN remain on a decline from period to period; on the other hand, the star similarity percentages display a slight rise in value (less than 2.5%) during several iterations of the network.

### 5.3.2 Trellis Similarity Analysis

The visualizations of the ASN showed a separation between the star-like points at the edge and the core of the network. The outer part of the core consists of low degree nodes that appear grid-like. The numbers for degree 3 and degree 4 nodes were fairly high and in proportion similar to the trellis generating function formula. In combination with the low count of degree 2 nodes outside the long stars, it helps support the presence of a pattern from the trellis ensemble within the ASN.

Table 5.2 shows the results for the dimensions for the dominant trellis in the network at each iteration, and Table 5.3 demonstrates the process used to find the dominant trellises. The $p$ and $q$ values represent height and width of the trellis mesh respectively. The first trellis observed was a 4x5 pattern. Roughly every twelve months, the trellis pattern became slightly larger as the width continually increased by 1. Rather
than using a different dominant trellis with a different $p$ value (a 5x6 trellis showed error less than 0.1) from the months before, the experiment was conducted with the 4x10 trellis for the final year of the study. The error in this adjustment did not alter the results much.

Figures 5.15 and 5.16 show the relationship between the dominant trellis terms

| Month | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
|-------|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| $p$   | 4 | 4 | 4 | 4 | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  |
| $q$   | 5 | 5 | 5 | 6 | 6  | 6  | 6  | 6  | 7  | 7  | 7  | 7  | 7  | 8  | 8  | 8  | 8  | 9  | 9  | 9  | 10 |

Table 5.2: The dominant trellis dimensions for the ASN based on error testing.

| Month | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
|-------|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| $p$   | 4 | 4 | 4 | 4 | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  |
| $q$   | 5 | 5 | 5 | 6 | 6  | 6  | 6  | 6  | 7  | 7  | 7  | 7  | 7  | 8  | 8  | 8  | 8  | 9  | 9  | 9  | 10 |

Table 5.3: The error testing for dominant trellis dimensions in each iteration of the ASN based on error testing. The $p$ values highlighted in light gray are the best fits while the values in dark gray are considered too far outside the range between 2 and the network diameter.
from the ASN and the trellis similarity for each month in the study. The ASN appeared to
go through two phases: the first phase in which it increased from 25% to 38% until the
midpoint of the study (month 30) and the second phase in which it remained steady
between 35% and 38%. The trellis similarity percentages are still below 40%. Side by

![Comparison of Dominant Trellis Components](image1)

**Fig. 5.15:** The distributions of the degree 3 and degree 4 nodes in the ASN.

![Trellis Similarity (2004-09)](image2)

**Fig. 5.16:** The calculated trellis similarity of the ASN to its closest trellis for each month of
the study.
side the two charts distinctly share a trend. The $x^3$ term of the ASN and the trellis similarity percentages follow almost exactly the same pattern over the course of the project.
Chapter 6

Conclusions

The Autonomous System Network is difficult to measure over time due to its large number of nodes, complex structure, and manner of evolution. This network is important to monitor though since it is the central routing system of the Internet. The need for a more robust and efficient configuration of autonomous systems becomes increasingly necessary. Researchers have sought to quantify and categorize the ASN in the past. While node counts, degree distributions, and other metrics have been amply noted in past studies, the properties of power-law have been argued. Attempts have been made to claim that the ASN is power-law, yet the final verdict on this routing network shows otherwise.

Generating functions have already seen use in the realm of examining real-world networks yet more applications could be achieved. The technique of Generating Function Comparison and Similarity algorithm (GFCS) extends their usefulness. This algorithm allows for a generating function based comparison of networks to familiar topologies. Four pattern ensembles in particular – fully-connected mesh, long star, chain, and trellis – were utilized with GFCS. The technique is proven to handle situations in which any number or combinations of patterns are prevalent in a network. Such combinations include both homogeneous and heterogeneous forests.
The GFCS algorithm has also shown the advantages and disadvantages of using generating functions in pattern analysis. Many different patterns, especially those with different generating functions, can be readily analyzed. Additionally, this pattern analysis method shows their accuracy and flexibility for this purpose. GFCS showed the changing trend in the ASN topology through pattern similarity despite the growth of the network and dominant patterns contained within it.

Their use in the study of the structure of a network or topology is not trivial however. Generating functions cannot pinpoint the location of a pattern within the topology or determine that the structure exists in one place. Pattern analysis through generating functions is only able to locate signatures of familiar patterns. The nodes belonging to any pattern could be scattered throughout the network without a visible pattern. Finally, several choices needed to be made in the process (use of the \( L^1 \)-norm, no inclusion of zero-coefficient terms in error testing, etc.) so this use of generating functions can be further improved.

The utility and potential for more applications of generating functions can be achieved. They have previously been used to create models and study various properties of real-world networks. This thesis has shown that they can also be used as formulas to study structures of real-world networks.

Through the use of this new technique and other existing metrics, several conclusions were drawn about the state of the evolving ASN. First of all, the numbers of autonomous systems and peer connections are steadily rising. Both are still growing linearly though the peer connectivity is increasing at a faster rate. Second, the ASN shows
increasing organization via clustering. The trend was constantly improving outside a four-month window between 2008 and 2009 when the ASN experienced a period of accelerated growth. Third, early iterations of the ASN showed many ASes with low or single degree indicating single-homed configuration. Regardless, the study reveals that the autonomous systems are continuously moving away from this single-home or star-like configuration and becoming more trellis-like. This transformation has been rapid from the very beginning of this study to the middle of 2006. Finally, there seems to be a fundamental transformation of the relative weights of edge and core ASN network where the ratio has shifted from roughly 70:30 to about 45:55 based on star and trellis similarity.

6.1 Future Work

The implementation of the comparison processes presented in this thesis could be utilized in several other ways. The method of topological analysis through generating functions does not have to be confined to the four suggested patterns. Many other topological patterns exist that could be used for comparison in this process. In addition, comparisons could be made between two different networks. The application of this process to the evolution of a network, testing the similarity of the same network at various intervals of time, is another prospect for future study.

The topological or pattern estimation in a large network can lead to exploration about various applied and engineering characteristics of the network. Star structures are normally not fault tolerant as they have a central point of failure, and a network with high star similarity may share that same vulnerability. On the other hand trellis structures suggest the existence of many paths between nodes, but the path to traverse a network
can be long. Networks that show high trellis similarity could show this inefficiency but remain robust while the opposite may be true as well. Many such questions could be posed about the significance of this new information.
References


http://lacnic.net/documentos/lacniciii/chapter-6-en.pdf
Appendix

Generating Function Comparison and Similarity Algorithm

GFCS(network graph G, network generating function G₀, set of pattern ensembles P) {
  1  set S to null set
  2  for each pattern ensemble c in P
  3    if G shows signatures of c
  4      for each term t in generating function formula Gᵢ of c
  5        if t is not in G₀
  6          continue with next pattern in P (break)
  7        add c to S
  8    sort patterns in S by ease of separability
  9  for each pattern ensemble d in S
 10    set m to maximum number of possible patterns in ensemble d
 11    for each individual pattern in ensemble d from i = 1 to m
 12      if the generating function formula for ensemble d shares an exponent with the generating function formula for any other ensemble in S
 13        set x to a term unique to ensemble d
 14        solve for all parameters of term x in i
 15        set n to the total number of nodes in pattern i
 16        subtract n from the number of degree x nodes in G
 17        set pattern generating function Gᵢ for i to generating function formula Gᵢظم
 18        for each term t with exponent k in Gᵢظم
 19          set coefficient of t equal to coefficient of term in G₀ with exponent k
 20        for each parameter p in Gᵢنظم
 21          solve for p and substitute the value into Gᵢ
 22    set Gₚ to sum of all Gᵢ
 23    set Δ to value that minimizes error between G₀ and Gₚ
 24    set d-similarity to 1 / Δ
 25  return all d-similarity values
}

• Gᵢ and Gᵢنظم is the generating function formula for pattern ensembles c and d respectively; examples of pattern ensembles include but are not limited to long stars, trellised, and chains
• Gᵢنظم(x) = f(p₁, p₂, ..., pⱼ) is the pattern generating function for a pattern graph p with j parameters
• Δ is the common multiplier for error testing in line 23 as in (6)