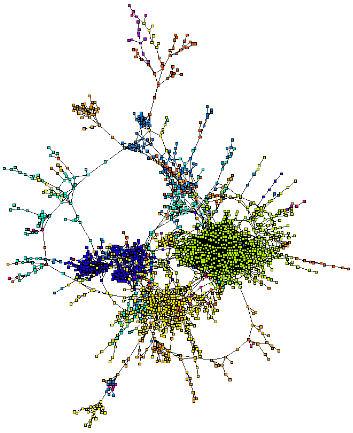
Fakultät für Informatik
Universität Karlsruhe

A Node's Perspective of Changing properties in Dynamic Networks

—An Evolving Model For The Internet at AS Level

Diplomarbeit
von
Lin Huang

(duckulareal@gmx.de)



December 2007

Ausgeführt unter der Leitung von Robert Görke

Referenten/Betreuer:
Prof. Dr. Dorothea Wagner
Dipl.-Math.-tech. Robert Görke
Fakultät für Informatik
Universität Karlsruhe

Declaration

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Karlsruhe, am January 17, 2008

Lin Huang

Abstract

A wide range of large-scale, rapidly evolving networks are omnipresent and of high importance in our daily life, such as communications, ecological system etc. Such networks are essentially large complex networks. In order to ensure the normal work of these systems, it is necessary to research these evolving networks. However, the structure and the feature of these different networks are still unclear to us. For decades the "Erdős-Rényi" random model has been a guide for the research until the recently proposed scale-free network. The research can be categorized into three areas: network topology generation and evolving model; the stability of the topology; the dynamics of the topology. In our work we focus on the evolving model for the Internet topology concerning the dynamics of nodes in the network over time, because traditional analysis just views the mass of all nodes together. In our work, we particularly scrutinize how specific nodes behave over time with respect to certain measures, asking for examples how volatile a node's movement in the core hierarchy is over time or whether there observable tendencies.

With the help of graph theory, we chose several different well-known metrics (the metrics here is not meant mathematically, but the charateristic in networks), such as degree, frequency, core and rich club connectivity etc. for our analysis. We observed converted trade information of *dm* chemist supermarket during Oct. 2004 and Oct. 2006 into a product-receipt-product network. We applied our metrics, statistic and visualization approaches in our analysis of the p-r-p network. We found that the degree distribution of the nodes in the *dm* p-r-p network has a steady structure over the observation. The trade of products is also stable and moreover the customers exhibit certain shopping habit. There are 13% of the products stay in the maximal shell all the time. Judging by the value of rich club connectivity, the core products (e.g. daily used stuffs) are always bought together. The nodes distribution in each shell is chaotic. Whether a new added product is distributed in a low shell or a how shell is not predictable, since it depends a lot on the own feature of the product. We compared our visualized results of the network with the statistic results from the database and we found that our chosen metrics and approaches are suitable for analyzing an evolving network concerning a node's dynamics in the network over time.

Since our chosen metrics and analysis approaches are feasible for the evolving networks, we then observed the Internet at the AS level during Apr. 2004 and Feb. 2006. In the observation, we found the Internet topology has also a steady structure because the degree distribution is similar over the whole time. Nodes with low degree or in low shell constitute the majority of the network. Most of the degree or shell changes are in the low degree or low shell group. The nodes, which have high degree and are in the maximal shell

stay quasi the same over the whole time. Half of the nodes stay in the original shell. Other nodes move up or down to their neighbor shells. 97% of the new added nodes are assigned with low degree, or in another word, if a new node enters the networks, it starts with a low degree with 97% probability. New nodes are firstly assigned in a low shell and they will either stay in the low shell area or move to a much higher shell area according to their own feature (e.g. a backbone AS starts with low shell and will eventually move into a very high shell as time goes by.) . The distribution proportion for different degrees in a shell always stays the same. We also found the number of nodes in each shell evolves in an approximately linear fashion. Besides, we found the value of rich club connectivity (against 1% rank) decreased 1 each year during the observation, but for a definite assumption, we need more observation data in the future.

Since we found the Internet topology at the AS level does have some regularities with respect of a node's behavior, we propose three relative successful different evolving models on the base of our analysis of the Internet. We propose a universal transformation matrix $\overline{M_p}$ to change the start state of an Internet topology to the state at next time point and to simulate the Internet topology at any time point by repeating our algorithms. We also programmed a simulator as a realization of our algorithm and evaluate our algorithm by comparing the simulated Internet topology at the AS level with the original one. Furthermore, we propose two refinements (*average window size* and *universal transformation matrices with step n*) for our algorithm to make our model more precise. Additionally, we propose another evolving model using curve fitting on the dynamics of nodes in the transformation matrix over time. We also estimated the theoretical runtime for the three successful models in each attempt of our documentation. At the very end we also evaluated proposed models (including all the models and refinements) by comparing them with the real Internet system.

Zusammenfassung

Eine große Auswahl von skalenfreien, sich rasch-entwickelnden Netzwerken sind allgegenwältig und wichtig in unserem täglichen Leben, wie Kommunikationen, ökologischem Systeme usw. Solche Netzwerke sind im Wesentlichen große komplex Netzwerke. Um die Arbeit dieser Systeme sicherzustellen, ist es notwendig eines komplexen Netzwerkes zu erforschen. Jedoch sind die Struktur und die Eigenschaften des komplex Netzwerkes oftmals noch zu unklar. Das ER Zufallsmodell ist für Dekaden ein Maßgabe fuer die Forschung gewessen bis vor kurzem das sogenannte akalenfreie komplexen Netzwerk vorgeschlagen wurde. Die Forschung des komplexen Netzwerken kann in drei Bereiche kategorisiert werden: Erzeugung und Modellierung der Entwicklung von Netzwerkstopologien; die Stabilität der Topologie; die Dynamik der Topologie. In unserer Arbeit konzentrieren wir auf das entwickelnde Modell für die Netzwerkstopologie über die Zeit.

Mit Hilfe der Graphentheorie wählten wir einige unterschiedliche weithin bekannte Maße, wie den Grad, die Frequenz, die Kernstruktur und die "Rich-Club-Konnectivität" usw. für unsere Analyse. Wir beobachteten umgewandelte gesammelte Verkaufszahlen des *dm* Drogerie Supermarkt aus dem Zeitraum von Okt. 2004 bis Okt. 2006 in einem Artikel-Bon-Artikel Netzwerk. Wir wendeten unsere Maße, Statistik und Visualisierung in unserer Analyse des Netzwerks an.

Wir entdeckten, dass die Gradverteilung der Knoten im *dm* Artikel-Bon-Artikel Netzwerk eine konstant Struktur über dem Zeitraum der Beobachtung hat. Der Handel der Produkte ist auch stabil und außdem haben die Kunden bestimmte Einkaufengewohnheit. Es gibt 13% die Produkte bleiben stets in der maximalen Kernschall. Durch den Wert der "Rich-Club-Konnectivität", wir entdeckten, dass die Kernprodukte (z.B. die täglich benutzten Materialien) immer zusammen gekauft werden. Die Knotenverteilung in der Kernschall ist chaotisch. Es is nicht veraussagbar, ob ein neu hinzufügt Knoten in die niedrigen Kernschall oder in die hohe Kernschall zugewiesen wird, da es von Eigenschaften des Produktes abhaengt ist. Wir verglichen unsere visualisierten Resultate des Netzwerkes mit den Statistikresultaten von der Datenbank und wir entdeckten, daß unsere gewählten Maß und Methoden für das Analysieren eines sich entwickelnden Netzwerkes bezüglich der Dynamik eines Knotens im Netzwerk über Zeit geeignet sind.

Da unsere gewaehlten Maße und Methode für die entwickelnden Netzwerke geeignet sind, beobachteten wir dann das Internet auf dem AS Level im Zeitraum von Apr. 2004 bis Feb. 2006. In der Beobachtung des Internets auf dem AS Level, entdeckten wir, dass die Internet-Topologie auch eine konstant Struktur haben, da die Gradverteilung über der Zeit ähnlich ist. Die Knoten mit niedrigem Grad oder im niedrigen Shell setzen die Mehrheit

des Netzwerks fest. Die meisten Grad- oder Kernschalländerungen spielen sich beim niedrigen Grad-oder Kernschallgruppe ab. Die Knoten, die hohen Grad haben und in der maximalen Kernschhall bleiben, sind quasi die selben Knoten über der Zeit. Die hälfte der Knoten bleiben in der ursprünglichen Kernschhell. Andere Knoten bewegen sich nach oben oder nach unten auf ihre Nachbarshell. Etwa 97% aller Knoten, die neu in das Netzwerk eingefügt werden, wird ein niedriger Grad zugewiesen. Neue Knoten werden zunächst einer niedrigen Kernschall zugewiesen und dann bleiben sie entweder in der niedrigen Kernschall oder ziehen auf eine viel höhere Kernschall entsprechend seiner eigenen Eigenschaften um (z.B. ein Backbone AS fängt mit niedriger Kernschall an und bewegt sich schließlich in eine sehr hohe Kernshall, wie Zeit vergeht). Die Verteilung der Anteile der unterschiedlichen Grade in einer der Kernschall bleibt quasi immer die selbe. Wir fanden auch heraus, dass die Entwicklung der Menge der Knoten in jeder Kernschall quasi linear sind. Außerdem fanden wir den Wert der "Rich-Club-Konnektivität" gegen Rank 1% verringerte 1 jedes Jahr waehrend der Beobachtung, aber fuer eine definitive Annahme, benoetigen wir Daten über ein längeren Zeitraum.

Da wir herausfanden, dass die Internet-Topologie einige Regelmäßigkeit hat, schlagen wir vor, drei relativ erfolgreiche unterschiedliche Entwicklungsmodelle anhand unserer Analyse des Internets. Wir schlagen eine universale Umwandlungsmatrix $\overline{M_p}$ vor, um den Anfangszustand einer Internet-Topologie in dem Zustand am folgenden Zeitpunkt umzuwandeln und die Internet-Topologie zu jedem beliebigen Zeitpunkt zu simulieren, indem wir unsere Algorithmen wiederholen. Wir programmierten auch einen Simulator als Realisierung unseres Algorithmus und werteten unseren Algorithmus aus, indem wir die simulierte Internet-Topologie auf dem AS level mit dem ursprünglichen verglichen. Außerdem schlagen wir zwei Verfeinerungen (*die durchschnittliche Fenstergröße* und *die universale Transformationsmatrizen*) vor, damit unser Algorithmus unser Modell exakter bildet. Überdies schlagen wir ein andere Entwicklungsmodell vor, die die Kurveanpassung auf die Dynamik der Knoten in der Transformationsmatrix über Zeit verwendet. Wir haben auch die theoretishe Laufzeit für die drei erfolgreiche Modelle in jedem Versuch unserer Arbeit geschätzt. Am Ende werteten wir auch die vorgeschlagene Modelle (inklusiv alle Modelle und Verfeinungen).

Acknowledgment

First of all, I am very grateful for my referee and mentor, Prof. Dr. Dorothea Wagner and Mr. Robert Görke, without whose help I wouldn't have come so far. Secondly, I appreciate it that I am allowed to use all the *dm* data and the Core Decomposition program from Mr. Marco Gärtler, which make my work convincing and easier.

Contents

List of Figures

Chapter 1

Motivation

A large complex network is becoming a hot spot in the research area. They occure in lots of fields, such as communication, neural networks, economics and management etc. Each node in the network stands for an element of the system and the connection between them indicates the interaction between two nodes. For example, in the social network, a node denotes a person, an organization or a country, and an edge indicates the social connection between them; in the genetics network of the life system, the node and the edges illustrate the chemical interaction between proteins, which can serve for signal transduction from the kernel of a cell to the outside.

In our normal life, a wide range of large-scale, rapidly evolving networks are omnipresent and of high importance: the Internet (FFF99), the electricity network (AANa), the air transportation network (AANb), the WWW network (AJB99), the E-mail network (BFNW04), the food chain network etc. They are all related to our daily life. The increasing dependence on these networks reveals a serious question: How on earth are these networks reliable? In the year 2000, the virus intruded the E-mail system of the British Parliament, which led to the paralysis of the system. In the same year, the O'Hare airport was closed because of the storm in Chicago, which influenced the flight plans in the whole USA. In the year 2003, the collapse of the electricity network in the state California in USA let most New Yorker upset. And nowadays, the damage of the ecological system influences the living environment of ours. How can we prevent the spread of the virus on the Internet? How can we design a network (electricity, flight etc.) which is strong enough for dealing with unexpected trouble? How can we keep the balance of the ecological system? All the solutions to these problems are related to the research of complex networks.

However, the structure and the features of complex networks are still unclear to us. Firstly, the structure of complex networks is complicated. Often, there is still no precise idea about how two nodes are connected in the network. Besides, networks are developing. More and more new nodes are added into the network. The connections between two nodes increase too as time goes by. The research of complex networks can make use of graph theory to describe the evolving model, evolving regulation and functionality. According to the Erdős-Rényi (ER) model (ERm60), a realization of a random complex network was first proposed. The ER model has been for decades a guide for the research of complex networks. As the interest in complex networks grows, other scientists proposed the famous scale-free network. But until now, there is still no clear definition for a large group of complex networks.

Nowadays, the research of complex networks can be concentrated on the following three areas:

- Network topology generation model and evolving model. It simulates the real network according to the generation model.
- The stability of a complex network. The research focuses on the impact of the constraints on the structural features of the network, e.g. the ability of the network for bearing the attack.
- The dynamics of complex networks. It's the ultimate goal of the research on complex networks.

This documentation focuses on the evolving model for the network topology concerning nodes' dynamics in the network over time. Evolving models can capture the feature of the generation of the network and help to obtain the impact of microscopic processes on the network topology. In our work, we observed two different networks. Firstly, we analyzed the trade information network from *dm*, chemist supermarket, with our chosen metrics to verify the feasibility of our approaches applied to analyze evolving networks. And then we analyzed the Internet at the AS level and tried to uncover the hidden regularities and organizational principles of this evolving network. Furthermore, we proposed evolving models on the base of our analysis of the Internet. We propose a universal transformation matrix $\overline{M_p}$ to change the start state of an Internet topology to the state at the next time point and to simulate the Internet topology at any time point by repeating our algorithms. We also programmed a simulator as a realization of our algorithm and evaluate our algorithm by comparing the simulated Internet topology at the AS level with the original one. Furthermore, we propose two refinements for our algorithm to make our model more precise. Additionally, we propose another evolving model using curve fitting on the dynamics of nodes in the transformation matrix over time. We also evaluated proposed models by comparing them with the real Internet system.

This work is structured as follows:

The scale of the network in our real life is huge and the interaction between nodes is complicated, the topology of the network is still unclear. In the past, there were a lot of researches of the topology of a real system. In **Chapter 2** we will introduce the reader to some previous work that has been done and the results that have been achieved in the research of complex networks.

In **Chapter 3**,we will introduce some fundamentals like the different metrics (such as degree, frequency, core), the concepts (such as the concept in time series analysis), data source and tools (such as the R environment), we chose in order to accomplish the analysis and modeling.

How to get the analytic values according to the different metrics from the network graph will be introduced by listing the algorithms in **Chapter 4**.

In **Chapter 5** we will show you the results we have achieved by studying the *dm* receipt-product-receipt network and the Internet at the AS level. We found that both network topologies have a steady structure. But over time, they have their own regulations in the nodes' distribution, nodes' movement area etc., because of their different own features as different networks.

Since we have found some interesting points in chapter 5, in **Chapter 6** we are trying to model the Internet topology at the AS level by simulating the evolution of the Internet over the time in different ways.

In **Chapter 7** we summarize again what we have achieved in our study and outline the interests that we can work on in the future.

Chapter 2

Previous Work

Because the scale of complex networks in our real life is huge and the interaction between nodes is complicated, the topology of networks is still largely unclear. In the past two hundred years, the research for describing the topology of a real system could be divided into three phases. In the first hundred years, scientists believed that the relationship between the elements in a system could be described with some regular structure, such as Euclidean network in a two dimensional space. From the late fifties to late nineties in the 20th. century, large scale networks with no clear design principle were described with simple random networks. The idea of random graphs dominated the research in the complex network area for about forty years. Only until recently, the scientists found that a number of real networks are neither regular networks, nor random networks. They are, however, the networks, which have different feature from the above mentioned two networks. The most famous networks are the small-world network (WS98) and the scale-free network (BA99). The discovery of these two kinds of networks leads to more interest in the research of complex network. In this chapter we will introduce the reader some previous work that has been done and the results that have been achieved in the research of complex networks.

2.1 Regular Network

For a long time, it was believed that the relationship between the elements in real-world system could be represented by some regular network, such as Euclidean network in a two dimensional space. The most used regular network is the circle network with N nodes. In this network, every node is only connected to the nearest K nodes and every node has the same degree and clustering coefficient.

2.2 Random Network Model

Random graph (ERm60) theory was first studied by the Hungarian mathematicians Pal Erdős and Alfred Rényi. They proposed the classical ER model. The definition of the ER model is as follows: in the graph, in which there are N nodes and $C_N^2 = \frac{N(N-1)}{2}$ edges, randomly, g edges were picked out to form a random network, $G_{N,g}$. There are $C^g_{\frac{N(N-1)}{2}}$ such networks, which are constructed with N nodes and g edges. The probability of every network is the same.

Another random network model, which is equivalent to the ER model, is the polynomial model. It is defined as follows: the number of the nodes N is fixed. We assume that any two nodes have the probability p to be connected. Such a network is denoted as $G_{N,p}$. As we can see, the number of the edges in the whole network is a variable, $p\frac{N(N-1)}{2}$. Let G_0 be the random network with nodes V_1, V_2, $\cdots$, V_N and g edges, according to the above mentioned process, with probability $P(G_0) = p^g(1-p)^{(\frac{N(N-1)}{2}-g)}$, we can obtain network G_0. If $g = pC_N^2$, model $G_{N,g}$ is equivalent to model $G_{N,p}$. It is very easy to get one model from the other one. Fig. 2.1 is an example with $N = 10$ isolated nodes. They are connected with probability p. It illustrates this process with different probabilities.

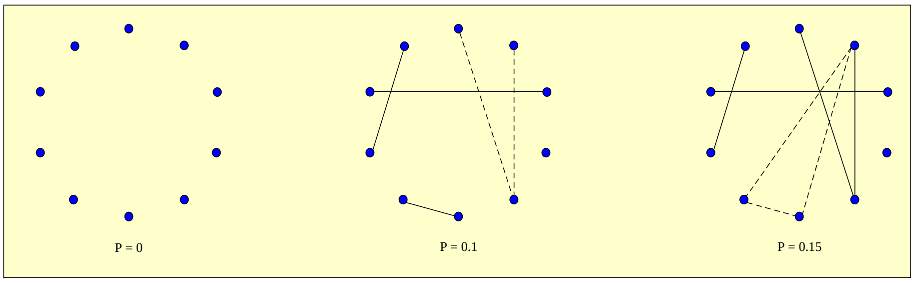


Figure 2.1: ER random network model

The degree distribution of the ER random graph is a Poisson distribution. It has a small average path length and a small clustering coefficient. From the late fifties to late nineties in the 20th. century, large scale networks with no exact design principle were described with simple random graph topology, in which the connections between nodes are random. In those years, some mathematicians researched the random graph and achieved a lot of approximate and even precise result through strict proofs. The idea of random graph controlled the research in the complex network area for about forty years. Only until recent years, because of the rapid development of the data processing and ability of computation by computers, scientists found that a number of real-world network are not completely random.

2.3 Small-world network

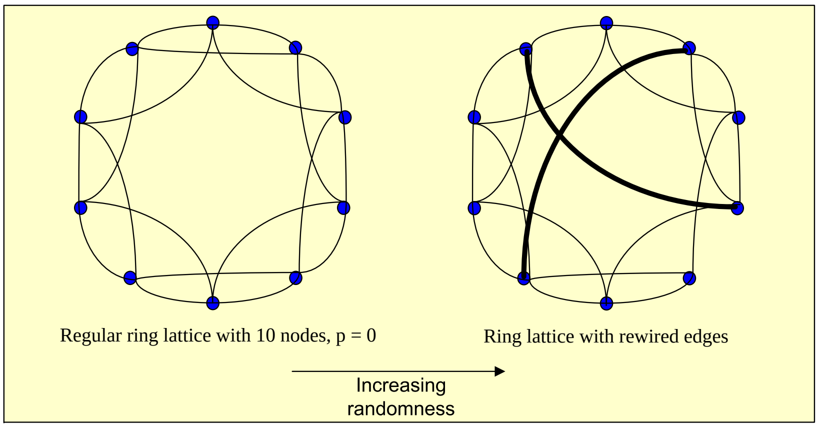

Regular ring lattice with 10 nodes, p = 0
Ring lattice with rewired edges
Increasing
randomness

Figure 2.2: Random rewiring procedure of the WS model

Through experiments, it is discovered that a lot of real-world networks, especially social networks, have a small-world character, which inspired the research of the small-world. The earliest small-world network model (WS model) (WS98) was proposed by Watts and Strogatz in the year 1998. It (see Fig. 2.2) starts with a ring lattice with N nodes, in which every node is connected with its m next neighbors on both sides. Every edge of the ring is then rewired with probability p (self-connections and duplicate edges are excluded). Those rewired edges are called ”long-range” edges, which decrease the average path length of the network and have less impact on the clustering coefficient in the network. The WS model has a social derivation: In the social system, most people know about their neighbors and colleagues and some of them have friends, who are far away from them, such as foreign friends.

After the WS model, Newman and Watts improved the WS model. They randomly add edges between any two nodes and rewire those new added edges. The original edges in the ring lattice stay the same. The improved model (NW) (NW99) is easier to analyze compared with the old one, because in the generation process of this model, there wouldn't be isolated clusters, which could happen in the old WS model. In the year 1999, Kasturirangan proposed a substitute (Kas99) for the WS model. It also starts with a ring lattice. And then new nodes are added to the lattice and connected randomly with nodes in the lattice. These randomly connected edges function as the ”long-range” edges.

2.4 BA Scale-Free Network Model

The degree distribution of the ER model and the WS model is actually a lot different from many real networks. There would be a lot of restrictions using random graph to describe these real networks. Therefore, some researchers tried to find another model to describe the networks better. In the year 1999, Barabasi and Albert researched the evolving process of the WWW network and discovered that most complex networks have such a feature that the degree distribution of the complex networks complies with a so called *Power Law.* And they referred this kind of networks as the scale-free network (BA99). Barabasi proposed that the incremental growth and the preferential attachment are two necessary mechanisms in the generation of the scale-free network. This thought has been widely accepted in the research area.

The original scale-free network model , which is called Barabasi-Albert (BA) model (or BA network) (BA99), is the first random scale-free network model. It starts with few isolated nodes (n_0 nodes) and periodically a new node with s ($s < n_0$) edges that link to s different existing in the graph nodes, is added to the graph. According to the preferential attachment, the BA model decides for each existing node v, whether it is connected to the new node s by the so called linear preference $L(v) = \frac{(d_v - c)}{\sum_j (d_j - c)}$, where j is an existing node in the graph, d is the degree of a node and c is a constant. As we can see, if an existing node has a high degree, then the probability $L(v)$ is high, which means that the new coming node is preferentially connected to this node. The probability is proportional to the degree of the existing node. We call this the preferential attachment. The average path length of the BA model is small and the clustering coefficient (FHJS02) is also small.

2.5 Deterministic Network Model

Stochasticity is the common feature of the small-world network and the Power-law complex network, namely, a new node connects to the existing nodes in the system according to different probabilities. However, as what was said by Barabasi (BRV), randomness is a major feature of most real-world networks, but it is hard to get a visual understanding of what makes them scale-free and how two different nodes are connected. Furthermore, the probability analysis and the random connection between edges are not suitable for the communication networks, whose nodes have fixed connectivity, such as the electricity network. Therefore, it would be of great interest to generate a small-world network or a scale-free network in a deterministic fashion. The advantage of a deterministic network is that we can analyze the features of the networks, such as the degree distribution, clustering coefficient and average path length, etc.

In the year 2000, making use of graph theory, Comellas et al. proposed a deterministic small-world communication network (CnP00). Two years later, they proposed two more models (CnP02) to generate the small-world network. In one network, every node has a constant connectivity and in the other not. There are also many other small-world network models. Corso from Brazil generates a small-world network (G.), in which the natural numbers are the nodes and they made use of the decomposition of natural numbers by primary numbers to decide the connections between nodes. With the help of number theory, Achter analyzed the main features of the network topology.

It was Barabasi, who simulate the first deterministic scale-free network. In the year 2002, Dorogovtsev et al. proposed a pseudofractal scale-free web (DGM02) generated by a simple structure mechanism and they analyzed the related network features, such as degree distribution, clustering coefficient and the spectrum of the network etc. In the year 2004, Comellas et al.(CGA04) determined recursive graphs with small-world scale-free properties. The best constructed deterministic network model should probably be the Deterministic Apollonian Networks (AJHAdS05). It was inspired by the problem of space-filling packing of spheres according to the ancient Greek mathematician Appollonius of Perga. This model is the scale-free and small-world features of the networks. The features of this network are similar to many real-world networks. It is widely applied.

2.6 Evolving Network Model

The idea of the BA (2.4) model provides others a new aspect of the research of complex networks. However, compared with many real-world networks, there seem to be apparent drawbacks in the BA model. As mentioned above, BA model has a small clustering coefficient, which is in real networks not true. And during the evolving process of the networks, each tiny change may have influence on the whole topology of the networks. Moreover, because of different influences (aging (DM00), competition (Bar00) etc.), different networks evolve very differently. Therefore, evolving network models attracted much interest in the research.

Preferential attachment is a widely accepted major mechanism in the Power-Law network. Nevertheless, many researchers proposed some other new mechanisms to generate the scale-free network. Kleinberg (KKR$^+$99) et al. and Kumar (KRR$^+$00) et al. proposed an evolving copy model to explain, how the Power-Law in the WWW network forms. Chung et al. suggested duplication models for biological networks (FYG03).

In real-world networks, the addition of new nodes, new edges, the removal of edges and the rewiring of the edges (AB00) are a set of basic events, which cause the evolution of the networks. Actually, any partial change in the network is caused by these four kinds of events or by the combination of them. BA model only considers the addition of new nodes. In the year 2000, Barabasi and Albert proposed an extended BA model (AB00). They researched the impact of the addition and rewiring of edges on the topology of the networks. In the BA model, old nodes always have a relative higher probability to get connected with new nodes. However, in the real-world networks, whether a node obtain a new connection is not only decided by its degree, but also by its competition ability (or fitness). Those node, which have a high fitness, may gain more connections than the node, which have a higher degree, but lower fitness. They will later become the nodes with high degree. This is called "the fitter are getting richer". To illustrate this phenomenon, Bianconi and Barabasi proposed a simple fitness model (Bar00). In this model, according to some certain distribution, every node is assigned a fitness value and they assume that the connection probability of an old node is proportional to its fitness and its degree.

Besides scale-freeness, a big clustering coefficient is another feature of real-world networks. In order to capture this phenomenon, Holme et al. , with the help of the mechanism "Triad Formation", proposed a scale-free network model (HK02), whose clustering coefficient is tunable by setting the probability of "Triad Formation". Inspired by citation networks, Klemm and Eguiluz proposed a highly clustered scale-free network model (KE02). This model divides nodes into two categories: active and non-active nodes. At any time, any two active nodes are connected. And if a new node is added, only an active node can be connected to it.

Above mentioned models are all 0-1 models, in which all the edges are the same, which is not true in the real life. Yook, Jeong and Barabasi were the first ones, who discussed the weighted evolving network model (YJBT01). They constructed a weighted network on the basis of an unweighted network, in which every edge is assigned a weight according to the relationship of the degree of every node. The research in the weighted network area is becoming more important. Especially, the proposal of the BBV model (BBV04) from Barrat, Barthelemy and Vespignani according to the statistic of the international flight network, inspired the research in this aspect. In the real life, the weight value is influenced by many elements. The investigation in weighted networks is a very difficult task and a big challenge.

Chapter 3

Fundamentals

In this chapter, we will introduce you to all the fundamentals that we use in this work, including different metrics, such as degree, frequency, core etc., that we used for the analysis. We will also talk about the tool R, we used to accomplish and present our analysis. Furthermore, we are going to introduce you to some general concept of the time series analysis that we use to make some simple analysis of our observation over the whole time.

3.1 The Definition and Illustration of a Network

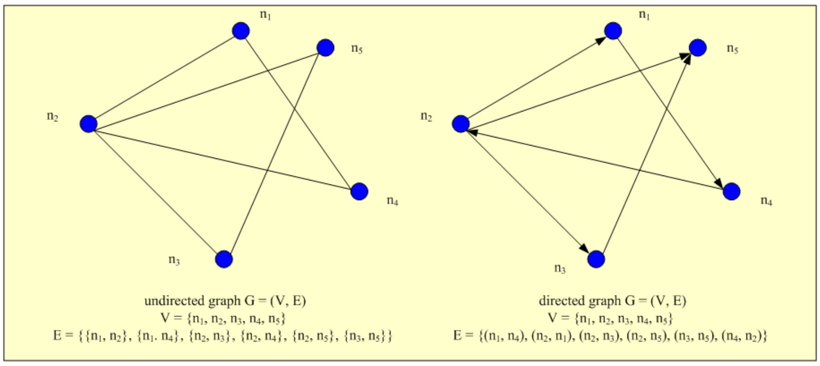

n_1
n_5
n_2
n_4
n_3
undirected graph G = (V, E)
V = {n_1, n_2, n_3, n_4, n_5}
E = {{n_1, n_2}, {n_1. n_4}, {n_2, n_3}, {n_2, n_4}, {n_2, n_5}, {n_3, n_5}}
n_1
n_5
n_2
n_4
n_3
directed graph G = (V, E)
V = {n_1, n_2, n_3, n_4, n_5}
E = {(n_1, n_4), (n_2, n_1), (n_2, n_3), (n_2, n_5), (n_3, n_5), (n_4, n_2)}

Figure 3.1: Undirected graph and directed graph

A network is presented as a graph mathematically. A complex network can be easily and precisely described by graph theory. A network (or graph) can be thought of as a set $G(V, E)$, where V and E are disjoint finite sets and we call V the vertex set and E the edge set of G. Every edge l_i in the set E has a related node pair u, v in the set V. If any node pair u, v and v, u in set E are the same edge, the network is an undirected network, otherwise it is a directed network. The number of nodes in the set V is called "order" and the number of edges in the set E is called "size". If order and size are finite, then the graph is a finite graph. The nodes connected by edges are referred to as end-vertices. An edge, which has the same end-vertices is a loop. Multi-edges or parallel edges are more than one edges that are incident to the same two vertices. A graph, which has no loop and no multi-edge is called a simple graph. The networks (or graphs) mentioned in this work are all undirected simple graphs with no multi-edge. The network topology is modeled by an undirected graph where the network devices are modeled by the nodes of the graph and the communication links are modeled by the edges of the graph.

3.2 The Internet at the AS level

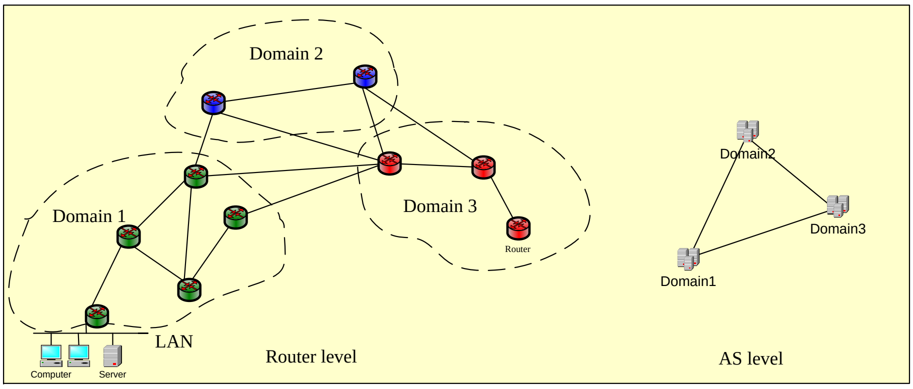
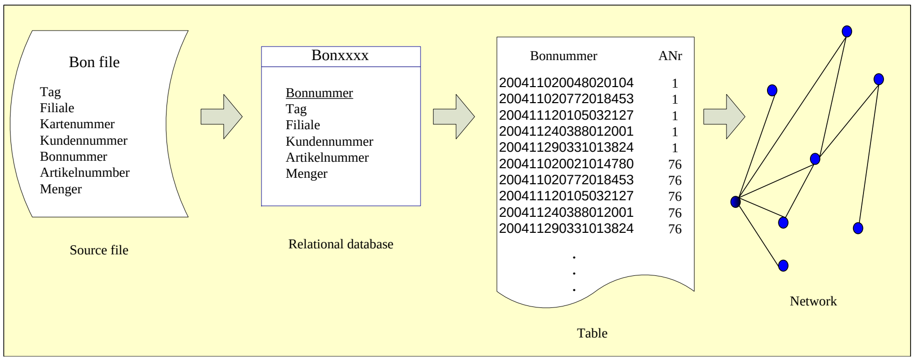


Figure 3.2: Internet topology at the router and the AS level

The Internet is composed of connected subnetworks which are known as domains or autonomous systems. They mainly consist of a large collection of routers and are under separate administrative authorities. Hence the study of Internet topology could be conducted at the router level, where each router is represented by a node or at the AS level, where each AS is represented by a node (see Fig. 3.2). In this work our study focuses at the AS level, since there are too many nodes (routers) at the router level. Furthermore, the AS level is particularly interesting, because they can be seen, technically, as internally homogeneous(composed of routers and have own polices), but they also form a network between them with routing policies etc.

3.3 Data Source

3.3.1 *dm* Trading Information



Bonnummer	ANr
200411020048020104	1
200411020772018453	1
200411120105032127	1
200411240388012001	1
200411290331013824	1
200411020021014780	76
200411020772018453	76
200411120105032127	76
200411240388012001	76
200411290331013824	76
. . .	

Figure 3.3: procedure from source data to network

The source data are about the trade information of customers in the chemist supermarket *dm*, who have pay-back cards. The data originally is all in CSV format. It includes four different kinds of files: Bon files, Kartennummer file, Atikelstamm file and Filialstamm file. For every month from Oct. 2004 to Oct. 2006 there is a so called Bon file.

1. **Bon files:** In these files each record is a receipt. It describes on which day, in which store, which customer using which payback card has bought how many and which products.

2. **Kartennummer file:** In this file the customer information is stored in the Kartennummer file. This file shows us the personal information of the card owner, where he always shops and besides *dm*, where he can use payback card to shop too.

3. **Artikelstamm file:** In this file each record describes the detail information of a single product, such as the number, the brand, import date of the product, etc.

4. **Filialstamm file:** Every record stands for the information of a store. Where is this store? When does it open? For questions like these, we all could look it up in this file.

To analyze these data, different kinds of network graphs were generated. We join the tables so that we get the relationship network graph. We could generate such a graph, where vertices stand for the products, and the line between two vertices means that these two products are together included

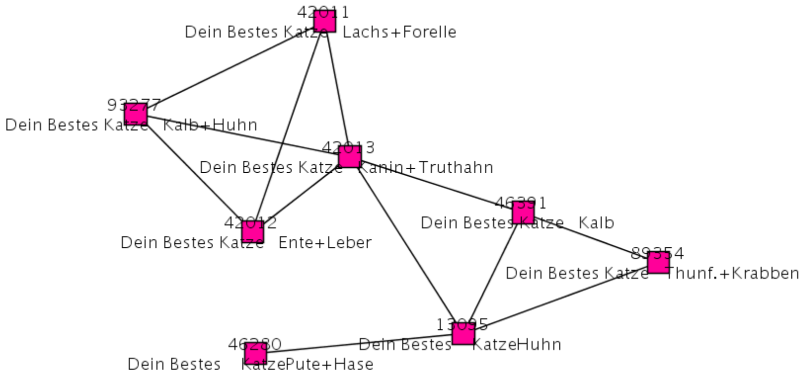

42011
Dein Bestes Katze Lachs+Forelle
93277
Dein Bestes Katze Kalb+Huhn
42013
Dein Bestes Katze Kanin+Truthahn
46391
Dein Bestes Katze Kalb
42012
Dein Bestes Katze Ente+Leber
89354
Dein Bestes Katze Thunf.+Krabben
13095
Dein Bestes KatzeHuhn
46280
Dein Bestes KatzePute+Hase

Figure 3.4: A small example of product-receipt-product network of *dm* trade information

in at least two receipts. A graph whose vertex is the product and the edge is the customer, stands for the fact, that the connected products were bought by at least two customers. Likewise, if a graph whose vertex is the customer and the edge is the product, it means that the connected customers have bought the same product. We can also get a graph with the receipts as the vertices and the products as the edges. The connected vertices denote that two receipts include at least two same products. In this way, we can discover the relationship among the products, customers and receipts. In this work, we analyze the product-receipt-product (3.4) network, in which a node stands for a product and if two products are two times bought together, namely, on the same receipt at least twice, then they have a connection, or an edge, between them. Since the trade information for all stores in the whole Germany is huge, we only choose the product-receipt-product network of a major store, who has the most turnover.

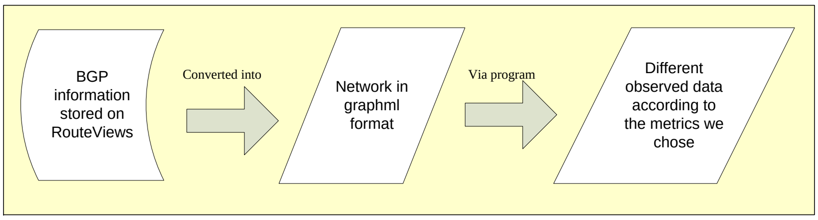

BGP information stored on RouteViews
Converted into
Network in graphml format
Via program
Different observed data according to the metrics we chose

Figure 3.5: procedure from source data to observed data

3.3.2 Route Views data

To analyze the Internet topology at the AS level. We use the oix-full-snapshot-xxxx-xx-xx-xxxx files, which can be obtained at www.routeviews.org and then converted them into graphml format. Such a file describes the Internet at the AS level at a certain time point. For example, oix-full-snapshot-2004-04-01-2200 describes the Internet at the AS level at 22 o'clock on Apr. 01 2004. The data in the files are described with XML as follows:

```
<graph id="G" edgedefault="directed">
  <node id="n0">
    <data key="d0" >
      <y:ShapeNode >
        <y:Geometry  x="-15.0" y="-15.0" width="30.0"
           height="30.0"/>
        <y:Fill color="#FF0000"  transparent="false"/>
        <y:BorderStyle type="line" width="1.0"
           color="#000000" />
        <y:Shape type="rectangle"/>
      </y:ShapeNode>
    </data>
    <data key="d1" >11537</data>
    <data key="d2" >true</data>
  </node>
```

It describes a node's id, geometrical position in a graph etc. Using "yfiles" to load and analyze the graphml file, we can compute the analysis information, such as degree of a node etc.

3.4 Metrics for a Complex Network

What we should notice here is that the metrics we speak in our documentation is a property, a characteristic or a measure in the network. It is not the concept of the metrics that we use in a mathematically rigorous way as usual.

3.4.1 Power-Law Metrics

Studying the properties of Internet topology actually consists of finding out the metrics that describe the properties. Such a topology model consists of several metrics, the nominal values for these metrics are measured according to the data of the real Internet topology. Within all the discovered Internet topology properties, f_d, the frequency of a degree d, is a basic foundation to judge if the topology graph is similar to the Internet topology. In the earlier studies, some researchers consider that the distribution of the degree of a node in the Internet is either totally random (Waxman model (BM99) or regular (Tiers (JCJ00)). But with the discovery of power laws (FFF99), it is proved that the Internet topology ranges between both of them.

In the year 1999, Faloutsos et al. analyzed the BGP information of the year 1998 from the National Lab for Applied Network Research (NLANR) and discovered that there are 3 Power-Laws in Internet topology (FFF99).

Power-Laws are expressions of the form $y \propto x^a$, where a is a specific constant of this law, x and y are the measures of interest and $\propto$ stands for "proportional to":

Power-Law 1 (rank exponent): The degree, d_v, of a node v, is proportional to the rank of the node, r_v, to the power of a constant, R: $d_v \propto r_v^R$;

Power-Law 2 (degree exponent): The frequency, f_d, of a degree, d, is proportional to the degree to the power of a constant, O: $f_d \propto d^O$;

Approximation (hop-plot exponent): The total number of pairs of nodes, $P(h)$, within h hops, is proportional to the number of hops to the power of a constant, H: $P(h) \propto h^H$, $h \ll \delta$, where δ is the diameter of the graph;

Power-Law 3 (eigen exponent): The eigenvalues, λ_i, of a graph are proportional to their order, i, to the power of a constant, E: $\lambda_i \propto i^E$.

In the topology graph, the number of the connections of a node with other nodes is referred to as the *degree* of a node. Every node in the graph with a certain degree has a *rank*. The higher the degree is, the higher the rank is. The *frequency* refers to the amount of nodes that have the same degree. The neighborhood size of a node n within h hops is the number of all the nodes that are reachable from node n within h hops from m. Furthermore, the pair size within h hops is the sum of neighborhood sizes of all nodes within h hops, it thus reflects the connectivity of a graph. The order i is the order of the eigenvalue, λ_i, in the decreasing sequence of eigenvalues.

Power-Law 1 implies that in the real Internet there is neither such total equality like in the Waxman model (BM99) nor a strict hierarchy like in Tiers (JCJ00) and Transit-Stub models. It suggests a "loose" hierarchy. Power-Laws 1 and 2 reflect that the actual Internet has the feature of high irregularity, which means that the minority has higher degree, while the majority has lower degree. For example, in the Internet, $R \cong -0.7$, $O \cong -2.2$ at the AS level and $R \cong -0.4$, $O \cong -2.4$ at the router level (FFF99). The exponent H in the approximation could be used to classify the topology graph. For instance, in the Internet, $H \cong 4.7$ at the AS level and $H \cong 2.8$ at the router level (FFF99). Power-Law 3 is used to further distinguish two similar graphs of the same kind. An example value of constant E could approximately be -0.4 at the AS level and -0.1 at the router level (FFF99). All these examples of constant values derived from the experiment results on the real Internet (FFF99), which are by now of course slightly outdated.

In our work we will use metrics frequency and degree as one of our main metrics.

3.4.2 Core

The concept of *cores* was proposed by Seidman in the year 1982, when he was researching the structure of the network. We call a set of nodes a *k-core* (Alb06), if each node in the set is connected to at least k other nodes in the set. We call a set of nodes a *k-core-shell* (Alb06), if all the nodes in this set belong to the *k-core* and do not belong to the $(k+1)$-*core*. To get the *cores* of a graph, the following steps are repeated:

1. Put all nodes with degree i (starting with i=0) into *shell* i
2. Remove them from the graph
3. Search again repeatedly in the new graph until no more nodes with degree i are found.
4. Increase i by one.

Fig. 3.6 is an example of k-cores and k-core-shells. All nodes are in the core 0. The second core (core 1) includes all nodes except the isolated gray node. The core 2 is composed of green and blue nodes. Finally, the blue nodes constitute the core 3. Concerning the core-shell, the number of the nodes in shell 0 is 1 because only the gray node in core 0 and not in core

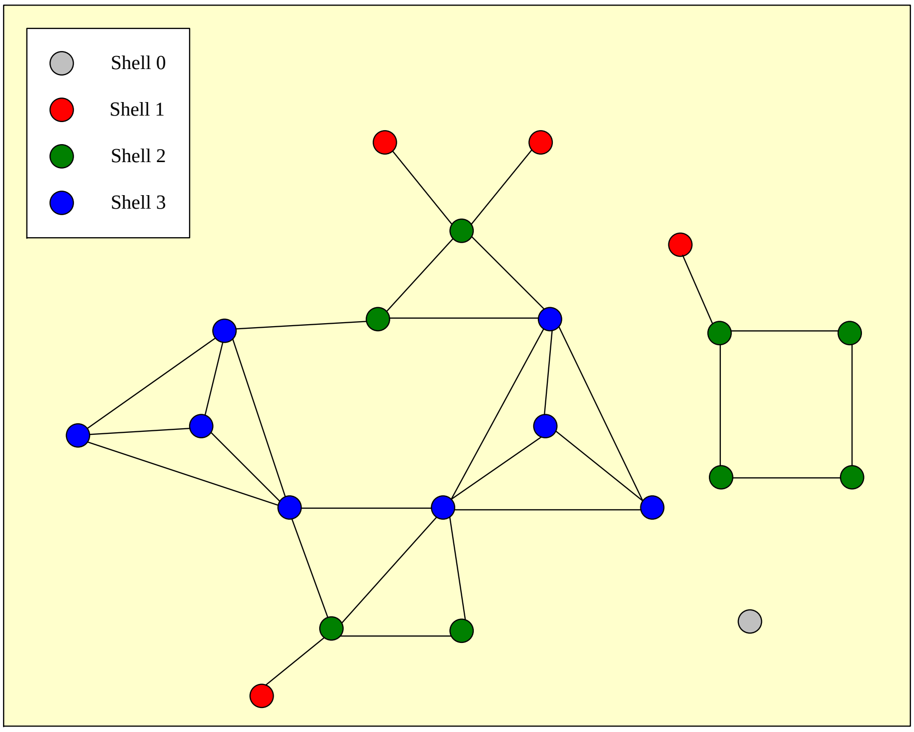


Figure 3.6: k-Shell Structure

1 belongs to shell 0. Accordingly, the red nodes are the elements of shell 1, the green ones belong to shell 2 and the blue ones are the nodes in shell 3.

As we can see, there is some relationship between the core and the degree of a graph. A node with a low degree definitely doesn't belong to a core with a high order. However, a node with a high degree could belong to a core with low order. For instance, a node, which works as a provider, could have an infinite degree (star structure connections), but it is in a very low core. Actually, the real Internet network has in many cases such a feature of the core structure. Let's take a look back at Fig. 3.2. The AS graph has some interesting feature, such as its core structure. $70 - 85\%$ of the nodes are in core 1 and core 2 but not in the core 3. The cores with higher order have fewer nodes. However the maximum core number ($k \cong 26$ with 20k nodes in the real Internet) again is very large.

In this work, we define which shell a node belongs to as the concept of ***shellness*** of a node.

3.4.3 Rich Club Connectivity

The metrics degree and frequency in Power-Law focus on the connections of a node. The metric core focuses on the whole hierarchical structure of a network. But the metric rich club connectivity, we are also going to use, focuses on the connections inside a small-world group. As we read in Section 3.4.1, every node in the network has a rank according to its degree. Those nodes, which have the same degree, are assigned with a position arbitrarily in that group. Therefore, each node has a distinct rank. The nodes in the network are sorted decreasingly, which means, who has the highest degree has the highest rank (rank 1). We denote the rank as r and it is normalized by the total number of nodes N. Now we define the rich club connectivity $\phi(r)$ as follows: the rich club connectivity of a rank r is the ratio of the total actual number of links of the nodes, whose rank is higher than r, to the numbers of the links if these nodes are completely connected, namely,

$$\phi(r) = \frac{\Sigma connections\ of\ nodes\ within\ the\ rank\ r}{\frac{n(n-1)}{2}}$$

Through this metric, we acknowledge that the nodes with a higher degree (or a higher rank) are well connected between each other.

3.5 R

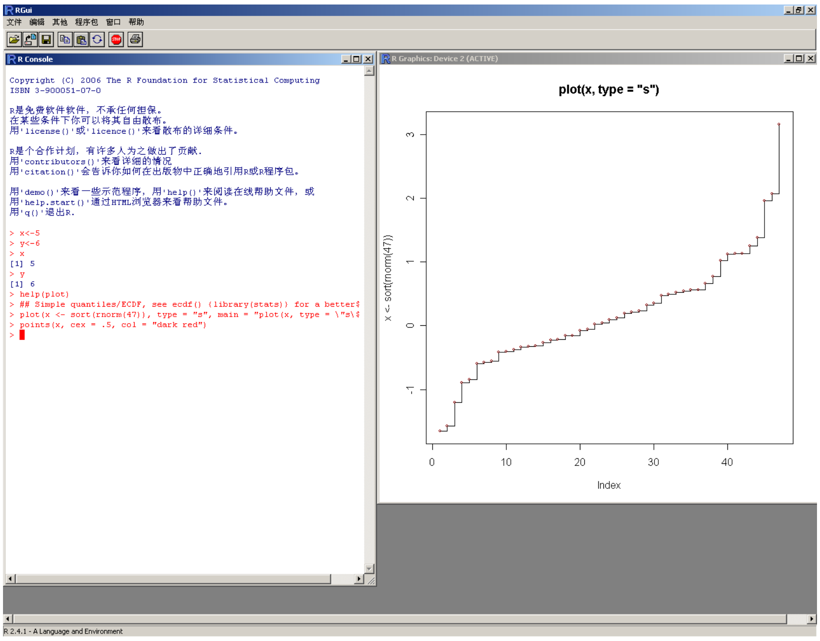
3.5.1 Benefits of R

R (VS) is a programming environment for data analysis and graphics. Most analysis in this work is accomplished with R. The following are the reasons why we choose R to analyze our data information:

1. R is a free statistics software. It is a copy of a commercial program named S. It's free, but its ability is not worse than any other same kind commercial software. According to its function, R and MATLAB are most alike.

2. R is an object oriented statistics programming language. It's easier for most of us, who are born in this OOP year, to understand and use R.

3. R has interfaces for other programming languages, such as JRE for JAVA. So that we can also call R functions in our java program.



RGui
文件 编辑 其他 程序包 窗口 帮助
R Console
R Graphics: Device 2 (ACTIVE)
plot(x, type = "s")
x <- sort(rnorm(47))
3
2
1
0
-1
0
10
20
30
40
Index
R 2.4.1 - A Language and Environment

```
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R是免费软件软件，不承任何担保。
在某些条件下你可以将其自由散布。
用'license()'或'licence()'来看散布的详细条件。

R是个合作计划，有许多人为之做出了贡献.
用'contributors()'来看详细的情况
用'citation()'会告诉你如何在出版物中正确地引用R或R程序包。

用'demo()'来看一些示范程序，用'help()'来阅读在线帮助文件，或
用'help.start()'通过HTML浏览器来看帮助文件。
用'q()'退出R.

> x<-5
> y<-6
> x
[1] 5
> y
[1] 6
> help(plot)
> ## Simple quantiles/ECDF, see ecdf() (library(stats)) for a better$
> plot(x <- sort(rnorm(47)), type = "s", main = "plot(x, type = \"s\$
> points(x, cex = .5, col = "dark red")
> 
```

Figure 3.7: Rgui

3.5.2 General Functions of R

R can accomplish some simple manipulation of numbers, vectors, arrays and matrices etc, such as assignment, arithmetic and comparison. Moreover, R can also realize the outer product of two arrays or the multiplication of matrices. One of R's significant functions is the statistics ability it provides. R has a set of statistics tables, which can evaluate and simulate the distribution of a data set, such as Poisson, binomial etc. Furthermore, it can generate a fitted statistical model for a data set.

3.5.3 Statistical Model of R: Linear Regression Model

The statistical model, we will use later (see 6.1), is a linear regression model. The data information over time might form a non linear curve. The linear regression model collects the analysis data together and fit them into a linear curve. Therefore, according to the fitted curve, we can predict the value of the data in the future. Suppose that the value of Y is influenced by the value of X_1, X_2 and X_3. The general form for Y would be $Y = f(X_1, X_2, X_3) + \varepsilon$, where f is the function that maps X to Y and ε is the error. Normally, we cannot figure out directly the f function. Therefore, we restrict it as the form $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$. Now, we only need to estimate the value of the sβs. R uses here the least square estimation. It defines that if the sum of the squared errors, $\varepsilon^T\varepsilon$, is minimal, then it estimates β the best.

$$\Sigma\varepsilon_i^2 = \varepsilon^T\varepsilon = (y - X\beta)^T(y - X\beta) = y^Ty - 2\beta X^Ty + \beta^TX^TX\beta$$

After the differentiation with respect to β and setting it to zero, we find the best estimation of β, $\widehat{\beta}$.

$$(y^Ty - 2\beta X^Ty + \beta^TX^TX\beta)' = 0$$

$\Longrightarrow$

$$X^TX\widehat{\beta} = X^Ty$$

$\Longrightarrow$

$$\widehat{\beta} = (X^TX)^{-1}X^Ty$$

For a simple regression $y_i = \alpha + \beta x_i + \varepsilon_i$,

$$\widehat{\beta} = \frac{\Sigma(x_i - \overline{x})y}{\Sigma(x_i - \overline{x}^2)}$$

The Gauss-Markov theorem proves that it is best linear unbiased estimate (see (Jul)).

3.5.4 Graphics of R

Besides the above mentioned versatile functionalities, R has also a very strong graphical facility. It can plot data sets with different options (two or three dimensional, multiple figures, piechart, histogram etc). For our linear model regression analysis, R can plot a series of residual summary (Fig. 3.8 is a random example) to show us whether the linear model is good or not. In the fitted value and residual figure, for each fitted value $\widehat{y_i}$ on X-axis, there is a corresponding residual value on Y-axis. Obviously, if the nodes are around the line $y = 0$, it means the linear model is good. The right upper figure is a normal quantile-quantile plot for residuals. The residuals are normal if this graph falls close to a straight line. The scale-location graph is a plot that calculates the square root of the standardized residuals. The highest points show us the largest residuals. The right bottom plot identifies points that have a lot of influence in the linear regression model.

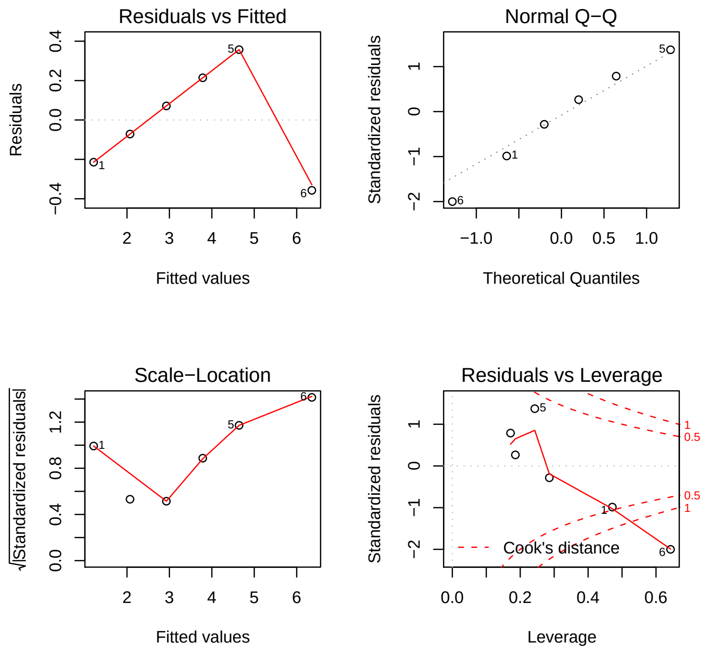


Figure 3.8: Example of residual plots

3.6 Time Series Analysis

3.6.1 Benefits of Time Series Analysis

Taking advantage of time series analysis, we can:

1. reveal the dynamic regulation of the development of the phenomenon we observe

2. forecast the future development

Time series analysis is used in many areas, such as sales forecasting, inventory studies etc. In order to observe different evolving networks, we also make use of some basic idea of the time series analysis, since the networks we study evolve over time. It helps us to understand the underlying structure that produced the observed data. And if we could find a proper model for the structure, it would be possible to forecast, monitor or simulate the networks.

3.6.2 Definition of Time Series

A time series is an ordered sequence of values of a variable at equally spaced time intervals. Two main factors constitute a time series. One is the time point. And the other is the value of the observed phenomenon at a time point. Normally we refer to the time point as t_i and to the observed value as a_i, where $i = 0, 1, 2, \cdots, n$. A time series can be written as $a_0, a_1, a_2, \cdots, a_n$ or $\{a\}$. We call a_i the *development level* at each corresponding time. We call the average value of a_i the *chronological average* or the *dynamic average*.

$$\overline{a}\ (dynamic\ average) = \frac{\frac{a_0}{2} + a_1 + a_2 \cdots + a_{n-1} + \frac{a_n}{2}}{n}$$

It denotes the average change of the observed value over time.

A time series can be categorized into absolute value time series, relative value time series and average value time series according to the observed value. Furthermore, it can also be categorized into time period series and time step series according to the time. In this work, we analyze the absolute value of our observation at each time step.

3.6.3 Development Statistics

We have the following standards that show us how the observed phenomenon develops:

1. increment amount $= a_n - a_0$, it denotes the total increment of the amount of the observed data;
   accumulated increment amount at each time point $=\{a_1 - a_0, a_2 - a_0, \cdots, a_n - a_0\}$, it is the increment of the amount at each time point compared with the one at the first time point;
   increment amount for each time step $= \{a_1 - a_0, a_2 - a_1, \cdots, a_n - a_{n-1}\}$, it describes the increment of amount of the observed data for a time point to the next;

2. development rate $(v) = \frac{a_i}{a_0}$, it denotes a development rate at each time point compared with the first time point;
   development rate for each time step $= \{\frac{a_1}{a_0}, \frac{a_2}{a_1}, \cdots, \frac{a_n}{a_{n-1}}\}$, it describes the development rate of the observed data for a time point to the next;

3. increment rate $= v - 1$, unlike the development rate, the increment rate denotes how fast the observed data increase;
   increment rate for each time step $= \{\frac{a_1 - a_0}{a_0}, \frac{a_2 - a_1}{a_1}, \cdots, \frac{a_n - a_{n-1}}{a_{n-1}}\}$, it

describes the increment rate of the observed data for a time point to the next;

4. geometric average development rate

$$\frac{a_1}{a_0} \cdot \frac{a_2}{a_1} \cdot \cdots \cdot \frac{a_n}{a_{n-1}} = \frac{a_n}{a_0}$$

$$\overline{v} \cdot \overline{v} \cdot \cdots \cdot \overline{v} = \frac{a_n}{a_0}$$

$\Longrightarrow$

$$\overline{v} = \sqrt[n]{\left(\frac{a_1}{a_0} \cdot \frac{a_2}{a_1} \cdot \cdots \cdot \frac{a_n}{a_{n-1}}\right)}$$

it calculates the average value of the development rate over a long time observation;

5. average increment rate $= \overline{v} - 1$, it calculates the average value of the increment rate over a long time observation

Chapter 4

Program Algorithms used for the Analysis

In this chapter, we will list the algorithms of the programs that we used to calculate the different metrics, such as degree, frequency, shell, rich club connectivity etc., used for our analysis. All the base data is stored in the mySQL server in our institute.

4.1 Degree

The number of the connections of a node with other nodes is referred to as the *degree* of a node. yFiles is an extensive Java class library that provides algorithms and components enabling the analysis, visualization and the automatic layout of graphs, diagrams, and networks. With the help of it, we can obtain a lot of features of a network in graphml format, such as degree, number of edges etc. The calculated degree for each node at each time point with the respective node's id is stored in the mySQL server.

label	degree	degree	...	degree
	200404012200	200404082200	...	200602232200
11964	2	2	...	Null
30403	1	Null	...	1
6366	2	2	...	2
⋮	⋮			

Table 4.1: The node-degree table in mySQL server

Algorithm 1: To obtain the degree of a node from the network and store it in the database

```
Data: Network in graphml format
Result: Degree of each node
for $i \leftarrow 1$ to (graph.allnodes).length do
    node← graph.allnode[i] ;
    degree← node.degree ; /* yfiles library */
    save node.degree in database;
```

4.2 Core

We call a set shell k, if all the nodes in this set belong to the k-core and not belong to the $(k+1)$-core. We define that a node has a shellness k, if it belongs to k shell. We used the Core Decomposition program by Marco Gaertler from our institute to get the shellness of each node. The following steps will be repeated until there is no node without a shellness in the network: We firstly search for a node which is not assigned with a shellness and has the smallest degree in the network. This smallest degree is then the current shellness we are looking for. Now we look for all the nodes which has no shellness and whose degree is smaller or equals the current shellness. If we find such a node, we assign it with the current shellness and then search for all its neighbors. All its neighbors' degree will be decreased by 1 and if the neighbor has no shellness and the degrees of the neighbor is smaller or equals the current shellness, it will be put into a candidate list. After all the nodes, whose degree is smaller or equals the current shellness, have a shellness, we repeat the above mentioned steps for all nodes in the candidate list until the list is empty.

label	shell20040401220	shell200404082200	...	shell200602232200
10764	8	8	...	8
1103	15	16	...	15
3333	22	22	...	25
⋮	⋮			

Table 4.2: The node-shell table in mySQL server

Algorithm 2: To obtain the core of a node from the network and store it in the database

```
Data: Network in graphml format
Result: Shellness of each node
while There is a node without shellness do
    level←the smallest degree of the nodes without shellness;
    list candidate;
    while There is a node n without shellness and n.degree⩽level do
        n.core←level;
        (n.neighbors).degree←(n.neighbors).degree-1;
        if a node nn ∈ n.neighbors and nn has no shellness and
        nn.degree⩽level then
            candidate.put(nn);
    while Candidate list is not empty do
        cn←candidate.get();
        if cn has no shellness and cn.degree⩽level then
            cn.core←level;
            (cn.neighbors).degree←(cn.neighbors).degree-1;
            if a node nn ∈ cn.neighbors and nn has no shellness and
            nn.degree⩽level then
                candidate.put(nn);
```

4.3 Rich Club Connectivity

We define the rich club connectivity $\phi(r)$ as follows: the rich club connectivity of a rank r is the ratio of the total actual number of links of the nodes, whose rank is higher than r, to the numbers of the links if these nodes are completely connected, namely,

$$\phi(r) = \frac{\Sigma connections\ of\ nodes\ within\ the\ rank\ r}{\frac{n(n-1)}{2}}$$

We firstly sort all nodes in descending order in the network. Since the rich club connectivity is about the connections between at least two nodes, the node with the highest degree doesn't have this value. And we set the value of this node as zero in the initialization. Now we repeat the following steps until each node has a rich club connectivity: We pick out a node in the sorted nodes list once a time, and look for its neighbors using yFiles class. And then we find all the neighbor nodes whose rank are higher than the selected nodes. These neighbor nodes and the selected nodes are stored in the *rnl* list. Now we can calculate the actual quantity of connections among the nodes in the *rnl* list and divide it by the amount of connections among these nodes if they are completely connected. The ratio is the rich club connectivity of the selected value. The values for 1% rank of nodes for the *dm* p-r-p network and Internet at the AS level over time are listed in table 5.1 and 5.2.

Algorithm 3: Computation of rich club connectivity against rank of nodes

```
Data: Network in graphml format
Result: Rich club connectivity against rank of nodes
sorted_nodes$\leftarrow$sort(graph.allnodes) ; /* according to nodes' degrees in descending order */
; /* Initialization */
list $l$;
$i \leftarrow 1$;
$n \leftarrow$ sorted_nodes[$i$];
edgenum $\leftarrow$ 0 ;
$l$.put($n$);
rcc $\leftarrow$ 0;
; /* Computation of Rich club connectivity (rcc) */
for $i \leftarrow 2$ to (graph.allnodes).length do
    $n \leftarrow$ sorted_node[$i$];
    $nl \leftarrow$ n.neighbors() ; /* function from yfiles */
    $rnl \leftarrow nl \cap l$ ; /* pick out the nodes whose rank is higher than the current node */
    $edgenum \leftarrow sizeof(rnl) + edgenum$ ; /* sum the number of edges among nodes in the rich group */
    $b \leftarrow \frac{i(i-1)}{2}$ ; /* number of edges if nodes are fully connected */
    $rcc \leftarrow \frac{edgenum}{b}$ ; /* calculation of the value of rich club connectivity (rcc) as described in the above text */
    $l$.put(n);
```

Chapter 5

Analysis

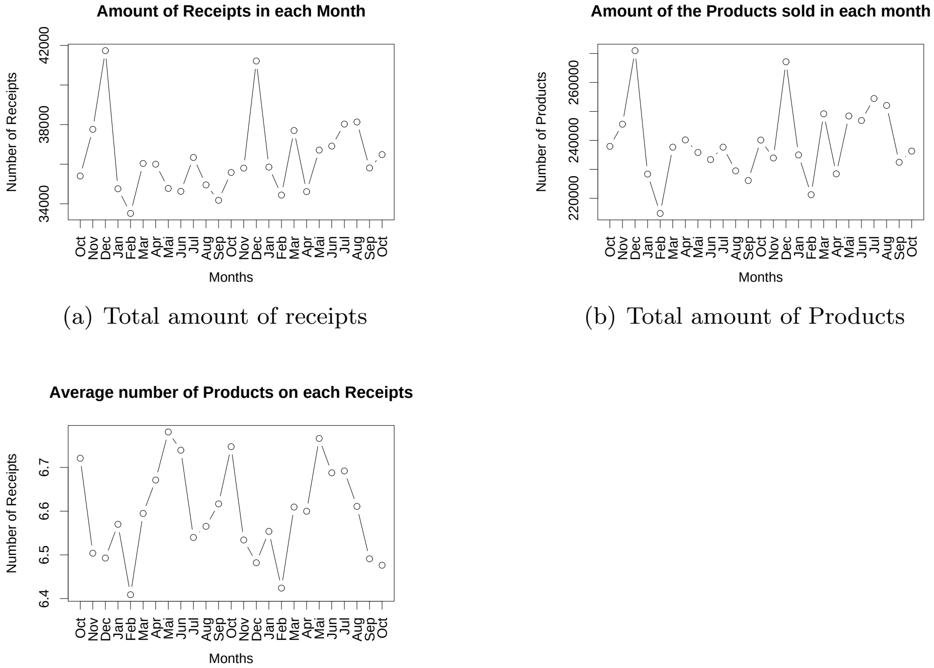
We obtained a lot of information from the chemist supermarket *dm* (see 3.3.1). We applied our approaches of the analysis to this real-world data to ensure the feasibility of the approaches and metrics (3.4) used for our analysis of an evolving network. We also obtained lots of information data from the Internet at the AS level (see 3.3.2). Applying our approaches that we approved in the analysis of the p-r-p network of *dm* chemist supermarket, we tried to reveal the hidden regularities of the changes of nodes as the Internet evolves over time.

5.1 *dm* Trading Network

We applied our approaches of the analysis to these sets of real data to ensure the feasibility of the approaches and metrics (3.4) used for our analysis of an evolving network. We should notice, that all our figures listed in our work are visualizations of our statistics and most conjectures we made in our analysis are proven facts, such as the degree/frequency distribution over time (5.3, 5.7), the movement of nodes among shells over time (5.12). We examined them by the statistic of the *dm* and the real Internet data in our database. Meanwhile, a few statements we made are only conjectures, such as the implication of a certain shop habit of a customer (5.7).

5.1.1 Overview

First of all, we will talk about what our chosen metrics represent in our observation of *dm* product-receipt-product network (3.4). A node with a high shellness implies that it is in a group of products in which lots of products are always bought together. The product groups with high shellness are composed of daily used products. Meanwhile a node with high degree can only tell us that it is more often bought or popular, but it cannot imply that it is in a product group. This is true, because a node with high degree can also have a low shellness. For example, a node n has a high degree, say, 500, but they are all connected to nodes $(m_1, m_2, \cdots, m_{500})$ with degree 1, then its shellness is only 1. As we can imagine, this implies that it is always bought together individually with another product, such as product pair (n, m_1) or (n, m_{56}), but not bought altogether, such as a products group $(n, m_1, m_2, \cdots, m_{500})$. For example, product like Nivea Visage Cleansing Milk, Tempo, Balea Bodylotion etc. are in a relative high shell group, the shell 20 group. The mentioned products are always bought altogether. And the baby food Bebivita has a relative high degree (about 85) but its shellness is about 30. It is not in a product group. However, in our *dm* observation, there is no such node which has a high degree but a very low degree, vise versa. Therefore, in this observation a node with high degree or high shellness means it is popular. The value of the rich club connectivity implies how well the nodes with high degrees are connected with each other. It denotes if popular products might be bought together. How strong the group is, that is formed by the most well-connected products.



(a) Total amount of receipts

(b) Total amount of Products

(c) Amount of products per receipt

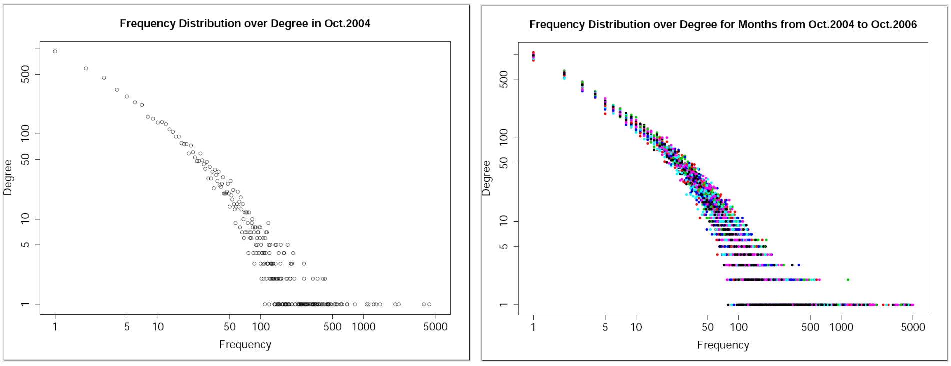
Figure 5.1: *dm* data Overview

Secondly, let's get an overview of the data we are observing. Fig. 5.1

are some figures that show us the general consuming states. Fig.(a) shows us how many receipts were printed out every month. Fig.(b) shows us how many products are sold out every month and Fig.(c) tells us how many products each receipt includes on the average. For example, in October 2004, there were about 238000 different kind of products sold out (same products sold out for more times are counted as one), about 37500 receipts are printed out and about every receipt includes about 6 products.

5.1.2 Degree-Frequency-Time

Fig. 5.2 are two figures that illustrate the degree distribution of the nodes of the network over the whole time. Fig. (a) is the degree distribution of nodes at one time point (Oct. 2004) and Fig. (b) is the degree distribution of nodes over the whole time. Different time points are illustrated with different colors. The x coordinate is the degree and the y coordinate is the number of nodes (or frequency) which have the same degree. The x and y coordinates are both in log form. As we can see, every month there are always about 900 nodes with degree 1 and about 600 nodes with degree 2 and the same is true for other degrees. This is an indicator for the network to have a steady structure over time.



Frequency Distribution over Degree in Oct.2004

Frequency Distribution over Degree for Months from Oct.2004 to Oct.2006

(a) Degree distribution of nodes in Oct 2004(other months similar)

(b) Degree distribution of nodes over the whole time

Figure 5.2: *dm* data Overview

Fig. 5.3 is a degree-time figure. Every node in the figure denotes a degree. The vertices (in the p-r-p network), which have the same degree, are illustrated by an identical node in this figure. Each line stands for the degree changes of a node as time goes by. As we can see, most nodes have few degree changes, which means even as the time changes, the trading of the products has a stability. Some items are always bought together. We can see in Figure 5.4, customers always buy the cat food (Dein Bestes) together over time. The most sold product is the non free plastic bag, which

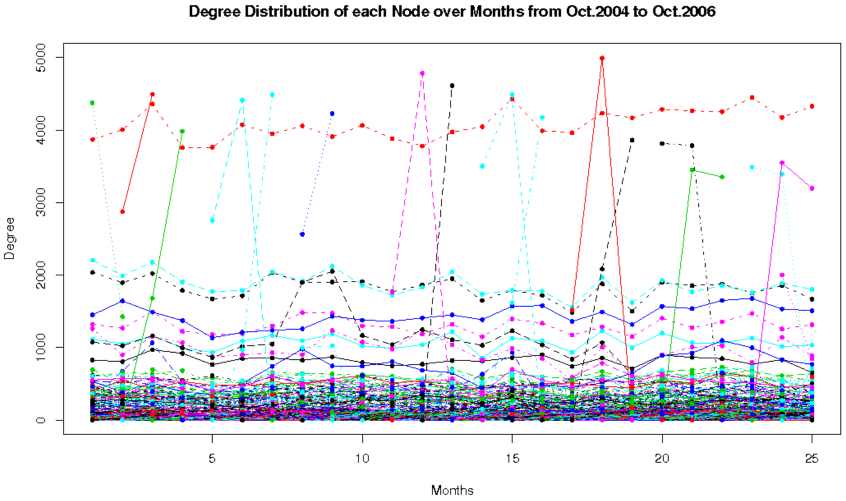
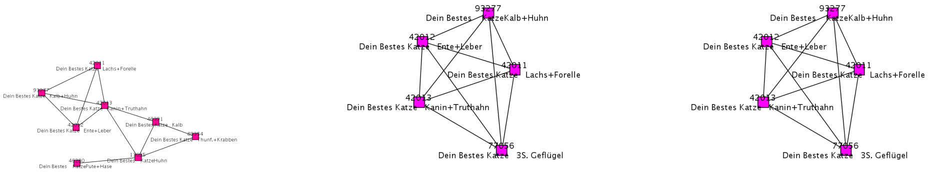


Figure 5.3: Degree of each node over time


93277
Dein Bestes KatzeKalb+Huhn
42012
Dein Bestes Katze Ente+Leber
42011
Dein Bestes Katze Lachs+Forelle
42013
Dein Bestes Katze Kanin+Truthahn
77056
Dein Bestes Katze 3S. Geflügel

(a) The p-r-p network for cat food (Dein Bestes) in Oct. 2004

(b) The p-r-p network for cat food (Dein Bestes) in Apr. 2005

(c) The p-r-p network for cat food (Dein Bestes) in Oct. 2005

Figure 5.4: An example of the stability of products group that is bought together

a customer uses to hold the products, he bought. Obviously, it is a popular product, which can be bought with any other products. The second and the third one are kitchen work and toilet paper. We can also find in the figure, there is a node, whose degree changes a lot over time. Sometimes it has the degree over 4000, and after couple of months it falls back to about 1500 or even lower. It is actually the payback card with a special bonus. Once a new payback card with new bonus is in the market, a lot of customers will buy it together with other products (lines with sharp peak) and after a period of time the card is sold out(the single red line on the very left in the figure). We can also see, that except for some special nodes, such as the node for payback card, each other node's degrees fluctuate around their original degrees as time goes by.

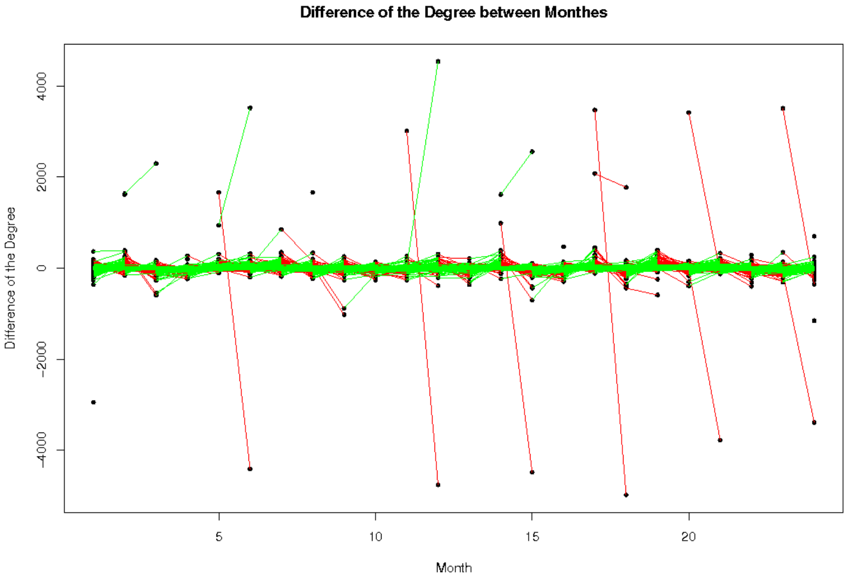


Figure 5.5: Difference of the degree between months

Figure 5.5 shows, how much the degree of a node changes from one month to the next month. The x coordinate indicates the step from one month to the next month and the y coordinate indicates from one month to the next month, how many degrees of a node have changed. The line between the nodes denotes the continuous changes through all months. For example, the single node on the left bottom in the figure means that the degree of this node from the first month to the second month decreases by about 3000.In this figure, we can find that the degree changes of most of the nodes ranges from −100 to 100. In this figure, we can also find, that except for some special nodes, such as the node for payback card, each other node's degrees fluctuate around their original degrees as time goes by, since the change alternates between being positive and negative.

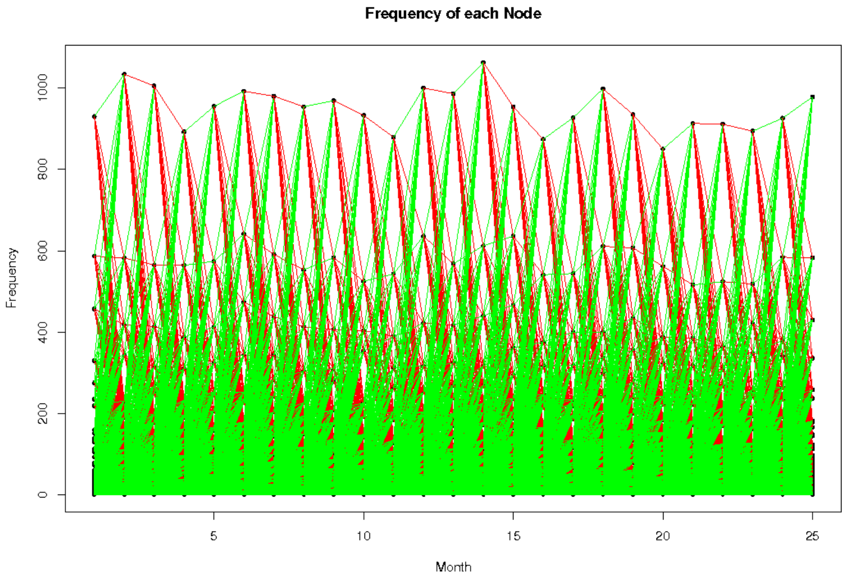


Figure 5.6: Frequency of each node over time

The following two figures are the frequency (of each node)-time figure 5.6 and frequency (of each degree)-time figure 5.7. As we can also see, the movement of the frequency, as the time changes, tends to be stable too. Every month, there are always about 1000 items sold with one other product, likewise, there are always about 600 items sold with two other products and so on. What is interesting, is that the distance (in frequency of each degree figure) between any two lines stays almost the same as the time passes. Meanwhile, in the frequency of each node figure, we acknowledge that every month there are some products that fall from a high frequency level to a lower frequency level, which means they have a higher degree than before, or in another word, they are sold together with more other products. And some of the nodes have the opposite situation. They get into a higher frequency level. Furthermore, a few nodes (or products) are not sold anymore. Because of these changes, the distance between any two lines stays stable. It indicates that all customers have a certain shopping habit. The customers always buy a lot of things, which are used commonly in our daily life, and the products, that are durable or rarely used are always sold less.

Figure 5.8 illustrates how many nodes have the same degree changes

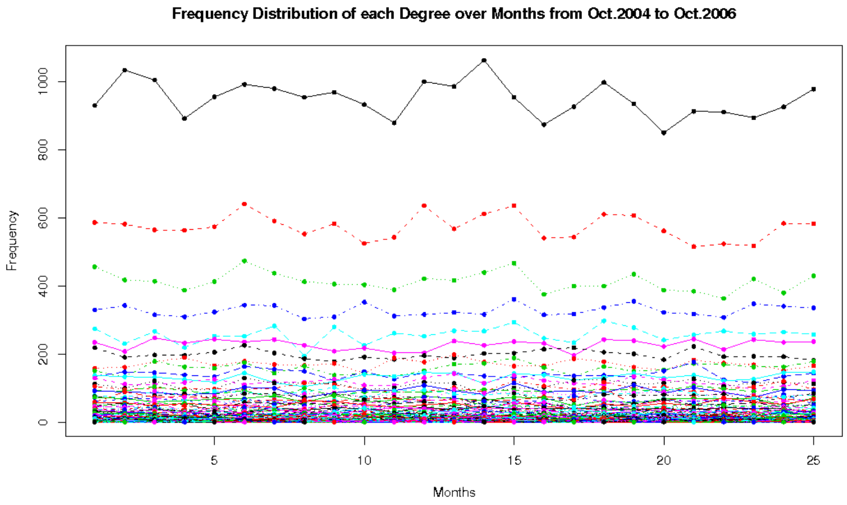


Figure 5.7: Frequency of each degree over time

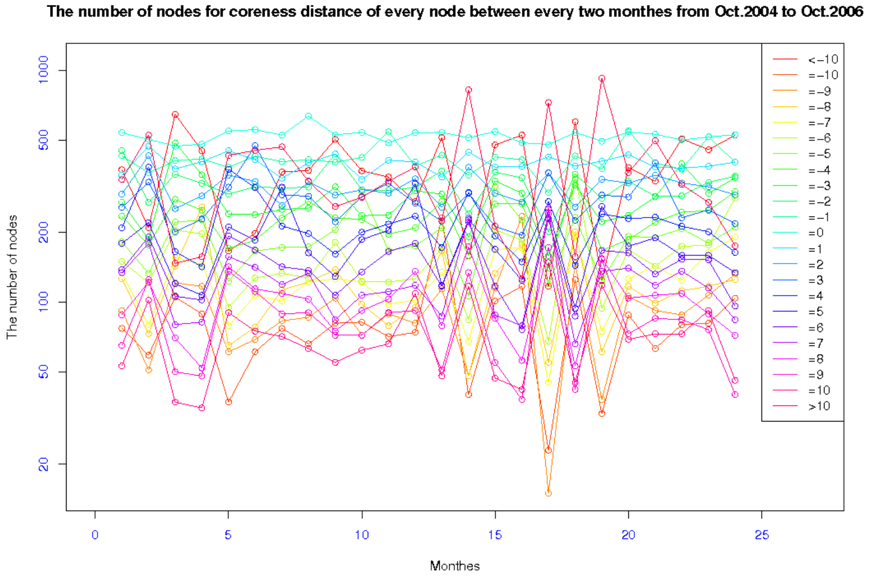


Figure 5.8: The amount of nodes for each degree change over time

between two months. The x coordinate indicates the step from one month to the next month and the y coordinate indicates the number of the nodes. Each line indicates, from one month to the next month how many nodes have the same degree changes. The different color of each line shows the range of the degree changes. For instance, the red line on the top means that from the first month to the second month, there are 800 nodes, whose degree changes greater than 10. And from the second to the third month, the degree of about 550 nodes changes over 10. The interesting discovery in this figure is that there are always two lines, whose movements are roughly the opposite. For example, the movement of green and yellow lines are opposite to the blue and purple lines. It is the reason that cause the overall stability of the degree and frequency changes in the above figures. The degree and frequency changes a lot, but macroscopically, they have stability.

5.1.3 Core-Time

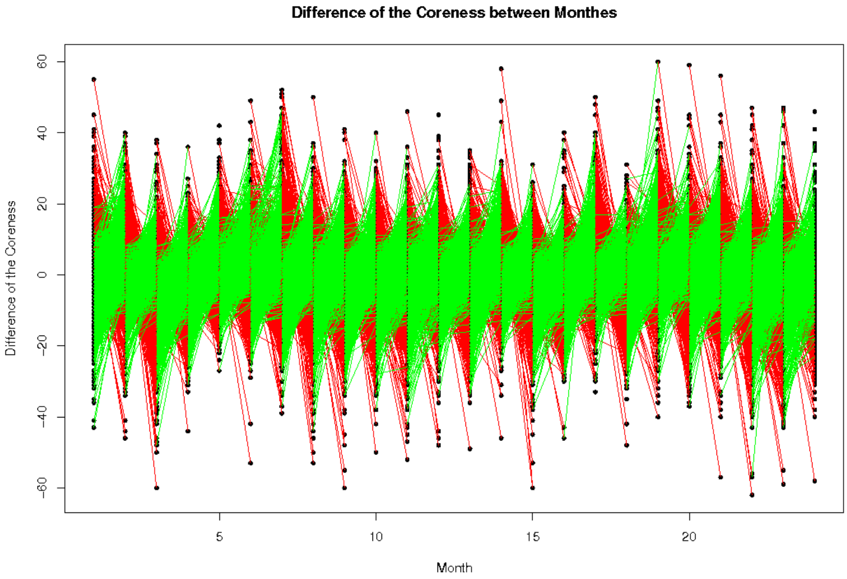
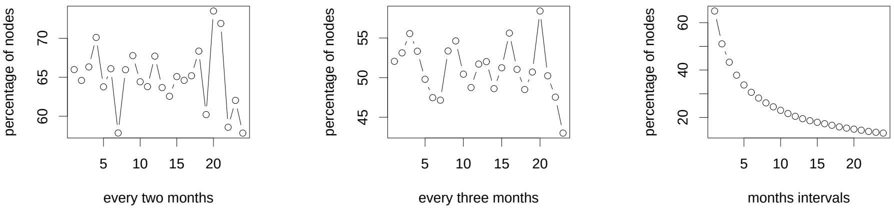


Figure 5.9: Differences of shellness between months

Figure 5.9 shows, how much the shellness of a node changes from one month to the next month. The x coordinate indicates the step from one month to the next month and the y coordinate indicates from one month to the next month, how many shellnesses of a node have changed. The line between the nodes denotes the continuous changes through all months. As we can obviously find that if the shellness of a node increases over about 12 in a month, it will not increase again. In another word, in the next step, its shellness will definitively reduce. The shellness of most products will not continuously increase over 12 in two months. Likewise, the shellness of most products will not continuously decrease over 12 in two months. That means, if a product is populary sold this month, it will maybe stay popular, but it is impossible, that it is becoming much more popular in the next month.



(a) Percentage of nodes which stay in the maximal shell every month

(b) Percentage of nodes which stay in the maximal shell every two month

(c) Average percentage of nodes which stay in the maximal shell over the whole time

Figure 5.10: Percentage of nodes which stay in the maximal shell over time

Figure 5.10 are some figures about the percentage of the nodes, which always stay in the maximal shell as time goes by. The x coordinate is the month interval. For example, in Fig.(a) the month interval is one (first month to second month, second to third and so on) and in Fig.(b) the interval is two (first month to third month, second to fourth and so on). Likewise, we can get a series figures with different month intervals (see the complete figures in appendix). These figures actually show us the overlap of nodes, which always stay in the maximal shell. For instance, in Fig.(a) in the first 15 months, there are always about 60% nodes stay in the maximal shell every month. In Fig.(b) from the third to the fifth month, there are about 55% nodes that stay in the maximal shell. Within all months of our observation, there are 13% nodes stay in the maximal shell. Fig.(c) is the average of all the figures. The x coordinate denotes the general month interval. '1' means the interval is 'every two months' and '2' means the interval is 'every three months'. The y coordinate indicates the percentage of nodes, which stay in the maximal shell. For instance, after 5 months (start month is random), there are 35% nodes still stay in the maximal shell. As we can see, the curve looks like a part of a normal distribution.

Figure 5.11 illustrates how many nodes has the same changes between two months. The x coordinate indicates the step from one month to the next month and the y coordinate indicates the number of the nodes. The

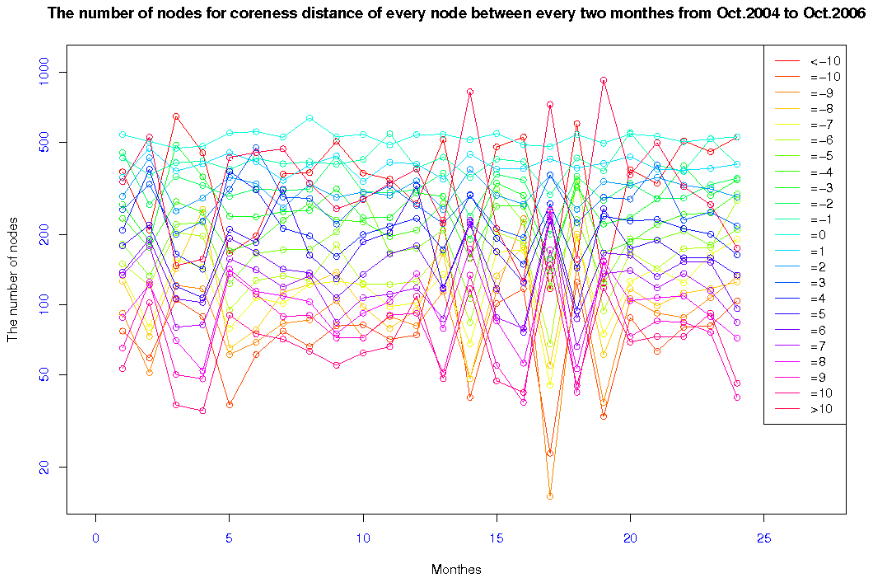


Figure 5.11: The amount of nodes for each shell changes over time

different color of each line shows the range of the shellness changes. For instance, the blue line on the top means, that from the first month to the second month, there are about 500 nodes, whose shellness changes are zero. It shows, every month, there are about 500 nodes, whose shellness does not change at all. The more changes of the shellness of a node, the less nodes there are. You may notice that the changes of the number of the nodes, whose shellnesses increase or decrease, are in correspondence. See month step 3 and 4, the total amount of nodes reduces, whose shellness increases by 1, 2, 3 or 4, meanwhile, the total amount of nodes increases, whose shellness decreases by 1, 2, 3 or 4. Actually, the movements of two lines, for instance, the number of the nodes, whose shellnesses increase by 1 and the number of nodes, whose shellnesses decrease by 1, are symmetrical. The sharp peeks at the month step 14, 17 and 19 are caused by the dramatic increments and then decreases of the shellness in those months.

Figure 5.12 is an illustration about how nodes move among all shells over time. The x coordinate denotes time stamps. The figure is composed of 25 pillars. There is a pillar at each time stamp. A pillar is filled with different color of short lines. Each line represents a node. The colors of the line illustrate different shells (Since there are too many shells, we use 26 different colors cyclically to represent different shells). The nodes in the first pillar are ordered increasingly by shells and illustrated by different colors.

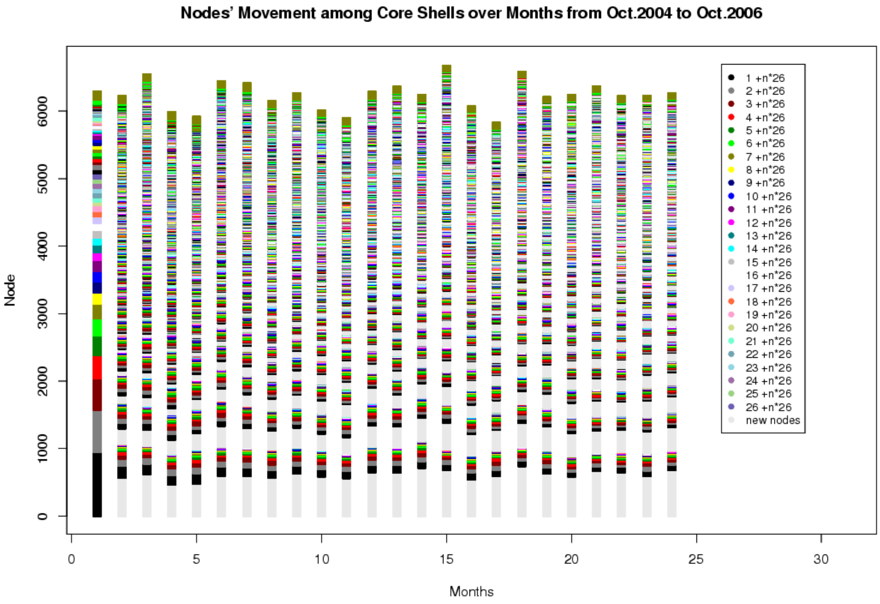


Figure 5.12: Nodes' movement among all shells

For example, black lines stand for shell 1 and dark gray ones stand for shell 2 and so on. The remaining 24 pillars are a little different than the first one. The colors of these pillars denote in which shell a node at the first time point was. And a node's current shell is illustrated by the shell tiers. A shell tier in our figures ends when the next shell, which the color represents, in the pillar is smaller than the current shell represented by the color. For instance, in the 25th pillar, shell 1 starts with the light gray color (new nodes) and ends before the next light gray color; shell 2 starts with the second light gray color and ends before the third light gray color and so on. The shellness of every node of these pillars at a time point is firstly ordered increasingly by its shellness and is then ordered by its original shellness at the first time point. For example, if a node was in the shell 1 at the first time point and is now in the shell 2 at the 25th time point, then it is in the second black area from the bottom at time point 25 in the figure. Unlike the Internet (see 5.21), the nodes' movement among the shells over the time is pretty random. New nodes could start in a high shell. A node which was in a very high shell could suddenly fall into a very low shell. Relatively, there are a few more nodes in the shell 1 and 2 than in other shells. But nodes are randomly distributed in all shells over the whole time. The growth of the quantity of the nodes is seasonal (decided by if the month is a consuming month,e.g. in December the amount of new nodes is more than other months.)

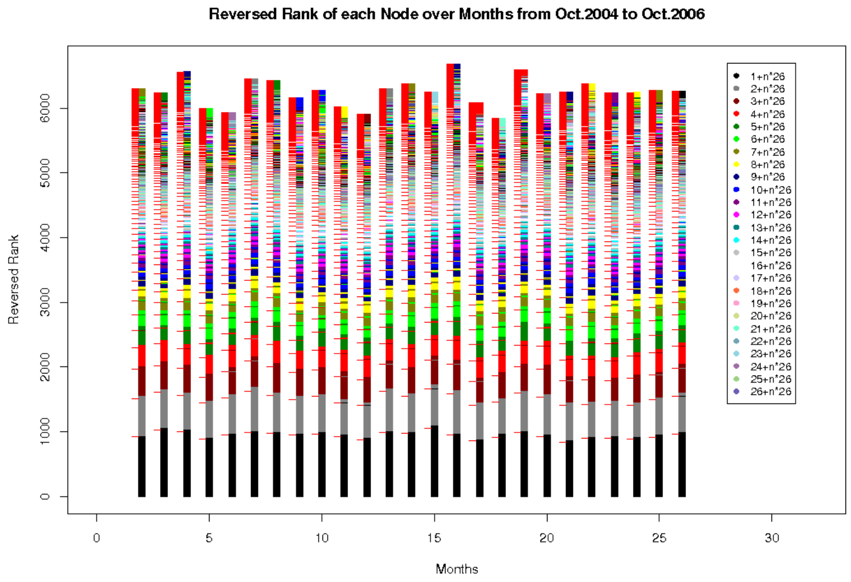


Figure 5.13: Distribution of nodes with different degrees in shells

Figure 5.13 illustrates the degree and shell relationship of the *dm* product-receipt-product network over the time. The x coordinate denotes the time stamps. The y coordinate is a node's degree ordered by its reversed rank. Every node has an identical rank as described in 3.4.1. A node which has the highest degree has rank 1. In this figure, for a better illustration, we reverse the rank. It means that the node with the smaller degree has smaller rank. The different colors here represent different shells (Since there are too many shells, we use 26 different colors cyclically to represent different shells). The red lines on the left of every pillars in the figure are the degree boundaries. For example, the area from the bottom to the first red line is for nodes with degree 1 and between the first and second line is for nodes with degree 2 and so on. Unlike the Internet (see 5.24), the nodes with different degrees are relatively average distributed in different shells. And if the amount of nodes in the network grows, the number of nodes in every shell increases. The growth of the quantity of the nodes in every shell is also seasonal. The maximal shell in every month changes.

5.1.4 Rich Club Connectivity-Time

Table 5.1 lists the rich club connectivity of the node whose rank is within the top 1% of all nodes. As it is said in 3.4.3, this metric tells us the connectivity inside a small-world group. The rich club connectivity is the ratio of the actual amount of connections among nodes and the quantity of links if these nodes were completely connected. The higher this value is, the better these nodes are connected. We chose 1% of the nodes which have the highest degree and calculated the rich club connectivity. As we can see, the connectivity among these nodes is around 80% over the whole time and always this high. It means most these 1% products are almost always bought together. It implies again that the core (general meaning, not the metric here) of the p-r-p network is stable and the customers have certain shopping habits. They always buy this stuff together.

5.1.5 Time Series Statistics

Over the whole time, there are on the average (dynamic average $\overline{a}$) 6260 nodes in the *dm* p-r-p network. From the start time to the end of the observation the geometric average development rate is 99.97%. The average amount of nodes in shell $1-10$ maximal shell are 985 and 7 and the respective geometric average development rates are 100.28% and 100%. The average highest shell is 56 and the geometric average development rate for the highest shell during the observation is 99.55%. There are average 5227 nodes in the low degree ($\leqslant$ 40) group and the geometric average development rate is 100.06%, meanwhile the respective values for the high degree ($\geqslant$ 200) group are 94 and 99.20% (see the complete statistics in appendix). Through the statistic we can see the total number of nodes, the amount of nodes in each shell, for each degree group changed little. The trade of the *dm* chemist supermarket was stable during the observation.

5.1.6 Summary

In this section, we converted the *dm* trade information into a general network (product-receipt-product network) and analyzed the network with metrics degree, frequency, shellness (3.4.1). In the analysis, we found that the degree distribution of nodes of the p-r-p network has a steady structure over time

Rich club connectivity $\phi(r = 1\%)$					
1-5 month	0.862599	0.848438	0.838987	0.829508	0.809040
6-10 month	0.800000	0.851442	0.859191	0.848710	0.861749
11-15 month	0.842938	0.831845	0.863942	0.796627	0.842406
15-20 month	0.839767	0.825248	0.820895	0.777778	0.861111
21-25 month	0.831250	0.856119	0.842262	0.847222	0.823413

Table 5.1: Rich club connectivity against rank r of p-r-p network

(5.2). The trade of products is also stable. Some certain products are always bought together (5.3). A popular product of a month wouldn't get sold more the next month (5.9). The customers have certain shopping habits. They always buy a lot of things, which are used commonly in our daily life, and the products that are durable or rarely used are always sold less (5.6). And they also have some certain consuming season, such as December and July are two best seller season meanwhile in February products are sold least (5.1). There are 13% products, such as non free plastic bags and kitchen paper, always stay in the maximal shell over time (5.10), which means they are always sold a lot together with other popular products over the whole time. The changes of the quantity of the nodes for each shell and each degree are symmetrical (5.8, 5.11) over the whole time, which means if some products are sold well, then there are some other products sold less. The core products, e.g. daily used stuffs, are always bought together, since they have a stable high rich club connectivity value (see table 5.1). It implies again that the core of the p-r-p network is steady and the customers have certain shopping habits. The nodes' distribution (including the new added nodes) in the shells is pretty chaotic (5.12): a node in a high shell could suddenly fall into a low shell, because it's sold less or a new added node could be in a very high shell at the first time, since it's popular. It depends on the own feature of the product. We compared our visualized results of the network with the statistic results from the database and we found that our chosen metrics and approaches are suitable for analyzing an evolving network concerning a node's dynamics in the network over time.

5.2 The Internet Topology at the AS Level

We observed the real Internet topology at the AS level every seven days from Apr.1 2004 to Feb.28 2006. We calculated the degree, the frequency, the shellness and the rich club connectivity of every node at each time point to accomplish our study.

5.2.1 Overview

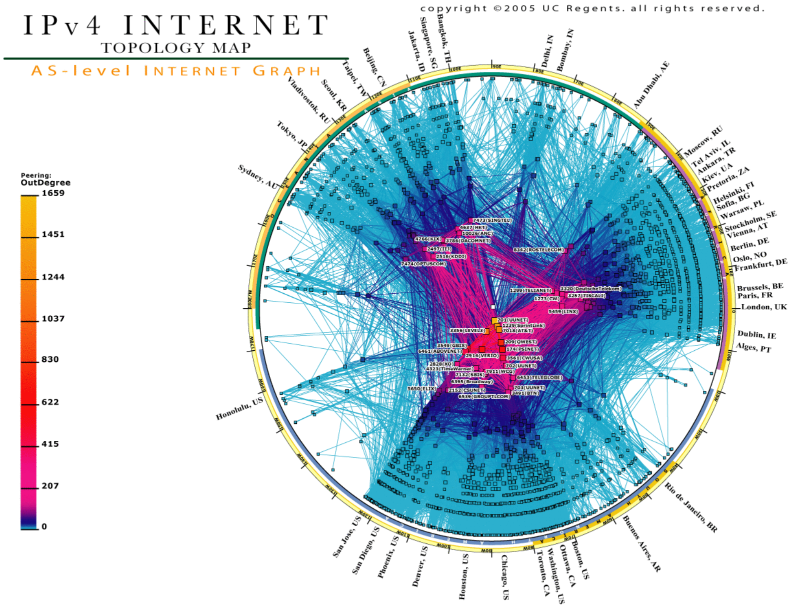


Figure 5.14: Internet topology at AS level

Here we will also firstly talk about what our chosen metrics represent in our observation of the Internet topology at the AS level. Figure 5.14 ((CAI)) is a macroscopic snapshot of the Internet for two weeks: 4 April 2005 - 17 April 2005 from CAIDA. The nodes are ordered by its degree. A node with high degree implies that there are a lot of other nodes connected to this node. Such node can be a big provider. As the node AS 701 (UUNET) in 5.14, which has the highest degree. A node with high shellness means it is in a node group, in which all nodes are well connected. Actually, most nodes with a high shellness are backbone ASes, such as UUNET and AS 3356 (Level 3 Communications). A node with high shellness could also have a low degree, e.g. AS 10310 (Yahoo) and AS 15169 (Google). Obviously, they are not big providers and they offer information retrieve system, which a well connection is very important. Meanwhile, a node with high degree could also have a low shellness, e.g. AS 8342 (company RTCOMM.RU), whose degree is 152 but its shellness is only 5. It implies it has a lot of clients (sub ASes), but its connection with the backbone ASes is weak. During our observation, it develops very fast. It connects to more backbone ASes and its shellness increased from 5 to 22. AS 721 is another example. It is an AS of Department of Defense in USA. It also has a high degree (149) but very low shellness (3). We assume, out of security reasons, it only connect to few but important backbone ASes. It has a high degree because it has a lot of sub ASes, which are inside the department.

5.2.2 Degree-Frequency-Time

Figure 5.15 is an illustration of the degree changes of every node in the real Internet topology over the time. Each line represents the degree change

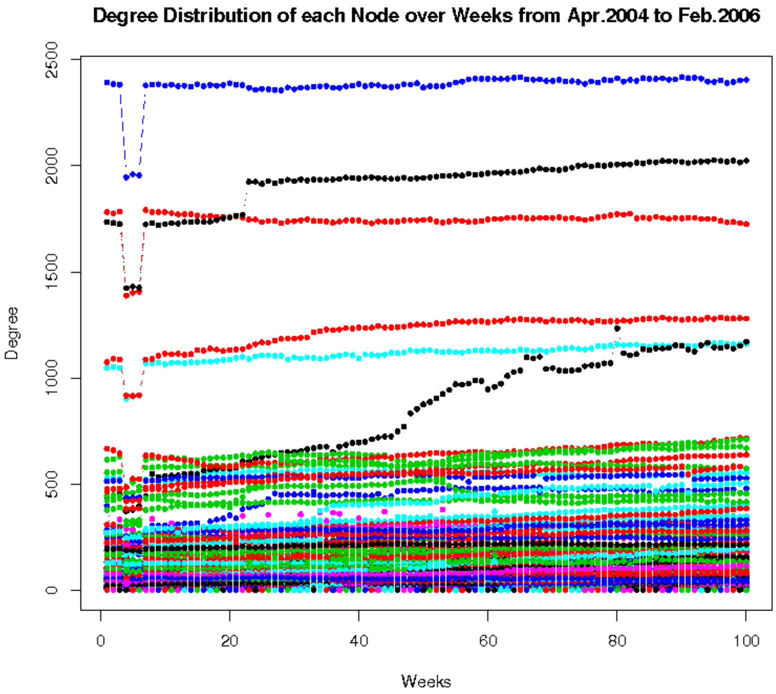


Figure 5.15: Degree changes of every node in the real Internet network topology over time

of a node. We can see that the maximal degree of a node is about 2,400 and most nodes have lower degree (under 500 and about 75% nodes have the degree 1 and 2). It complies with the Power-Law. The degrees of all nodes decreased dramatically in about April and May in the year 2004. The higher degree a node used to have, the more degree it lost in this period. It's actually a reflection of the preferential attachment theory proposed by Barabasi (BA99). When a new node is added into the network. It always tries to connect to a node with high degree. Once the system decreases and the nodes with low degree lose the connection to the nodes with high degree, it is obvious that the decrease rate of a node with high degree is bigger than others. During these two years, the degrees of most nodes increased slightly over time. One special node is the AS 174, whose degree developed from 429 to 1171. It's a multinational Tier 1 Internet service provider ranked as the largest Ethernet Service Provider, called Cogent. It offers different Internet access or transport services. The node with the highest degree is AS 701, which is also a provider.

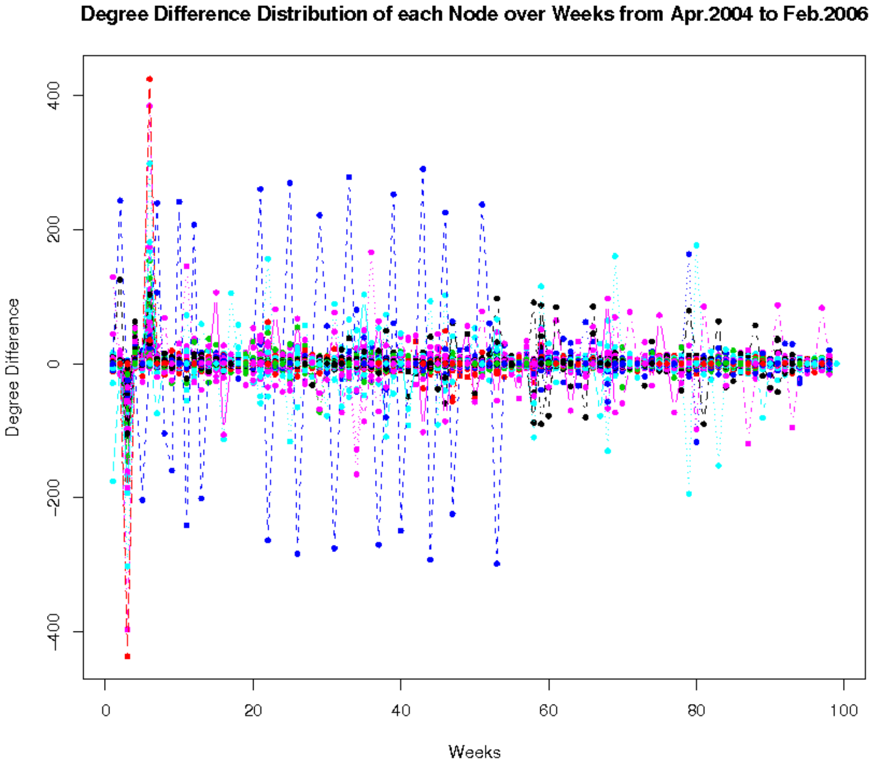


Figure 5.16: Degree changes of every node in the real Internet network topology over time

Figure 5.16 shows, how much the degree of a node changes from one week to the next week. The x coordinate indicates the step from one week to the next week and the y coordinate indicates from one week to the next week, how many degrees of a node have changed. The line between the nodes denotes the continuous changes through all months. Each line represents the changes of a node. For example, the red line, whose increment and decrease scope are over 200, is the degree changes of the AS 3246, which is assigned to a network operator. As we can see, most nodes have small degree changes (their degree differences are around zero). A few nodes have relative big variations, such as AS 3246. But if it increases 200 degrees this week, it will decrease 200 degree the next week. This kind of balance is the cause of the slight overall increment of the degree (see Fig. 5.15).

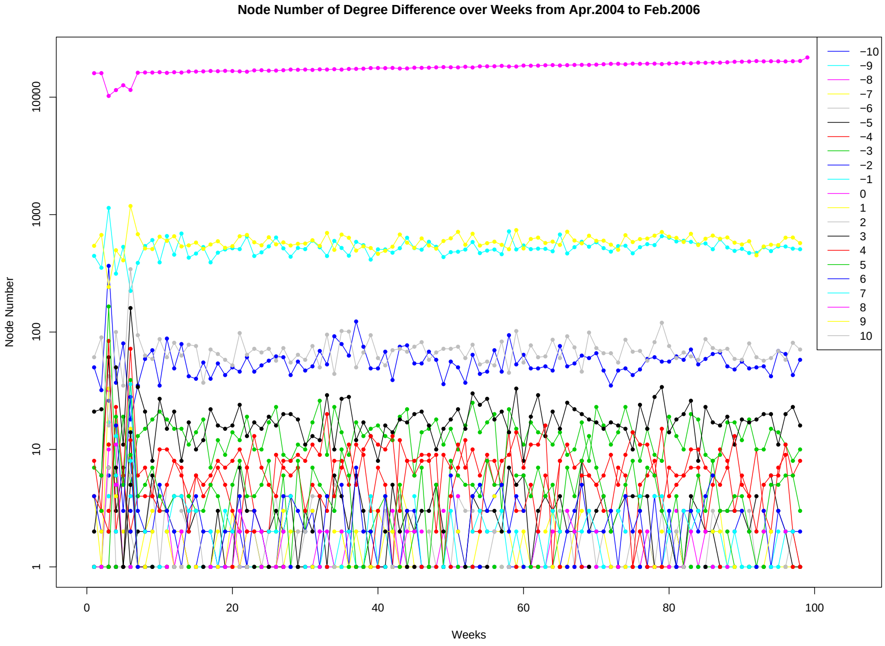


Figure 5.17: The amount of nodes with the same degree changes

Figure 5.17 is an illustration of the amount of the nodes which have the same degree changes at each time step. The x coordinate is the time step from one week to the next week and the y coordinate denotes, how many nodes have the same degree changes. Each line in this figure is colored to represent the degree changes. For instance, the pink line on the top represents no degree change and the yellow below it represents that the degree increased by one. It is obvious that over half of the nodes have no degree change at all over the whole time and most nodes have small degree changes (top 5 lines in the figure). Another interesting thing is that there are approximately same (or similar) amount of nodes by decrease and increase. For example, the yellow and light blue line around 800 in the figure. Respectively, they present the quantity of the nodes whose degree increase by 1 (yellow) and decrease by 1 (light blue). The values for each time step of two lines are very close. It is also true for the gray and blue and the other lines too. Furthermore, the percentage of nodes of the relative changes at each time step is very similar. These assure the stability of the Internet topology.

Figure 5.18 illustrates the amount of the nodes (frequency) which have the same degree. Each line in the figure denotes the frequency changes of a node group, which has the same degree. The red line on the top is the degree 2 group and the black line around 6000 is the degree 1 group. It complies with the Power-Law. The green and blue lines are the degree 3 and 4 group. Nodes with low degree constitutes the majority of the network. As we can see, most changes are in these four groups. The number of the nodes with high degree stays almost the same over the time. Actually, these nodes are the Tier 1 backbone ASes, their degrees change also few as time goes by. It assures the stability of the Internet topology.

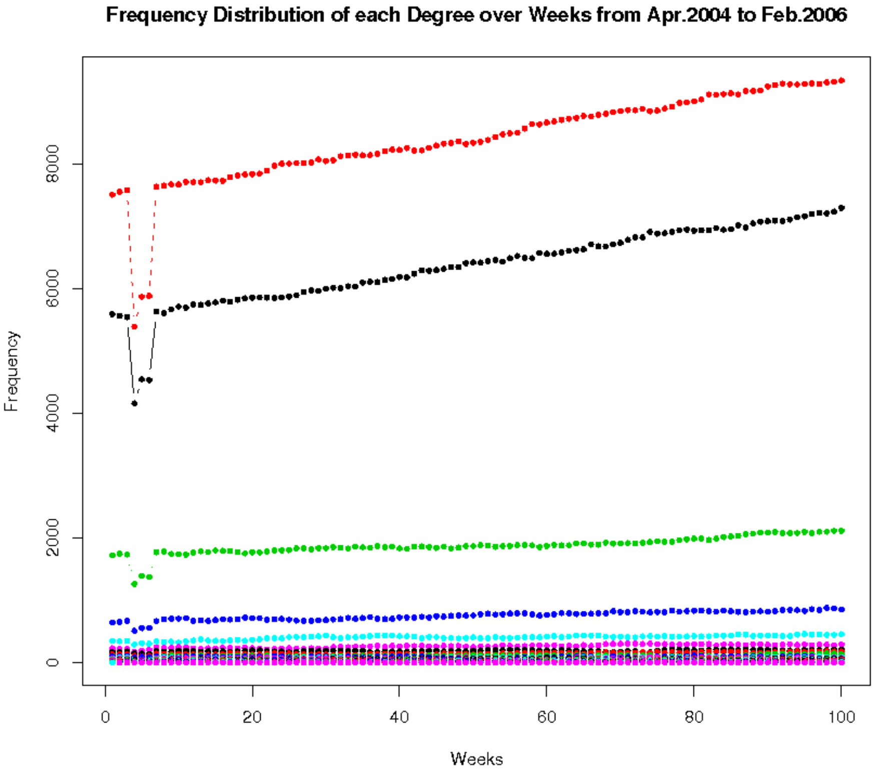


Figure 5.18: Frequency distribution of each degree in the real Internet network topology over time

5.2.3 Core-Degree-Time

Figure 5.19 shows us how much the shellness of a node changes from one week to the next week. The x coordinate indicates the step from one week to the next week and the y coordinate indicates from one week to the next week, how many shellnesses of a node have changed. The line between the nodes denotes the continuous changes through all weeks. Each line represents the changes of a node. As we can see, most nodes have small shellness changes (between -3 and 3). A few nodes have relatively big variations. For example, the pink line, whose decrease scope is over 20 at time step 36, is the shellness change of the AS 9942, which is assigned to APNIC (Asia Pacific Network Information Centre). Actually, from Apr. 2004 to Nov. 2004 AS 9942 always stayed at the maximal shell and from then it fell dramatically to the shell 5 and stayed there. Another example is the AS 25462, which is assigned to a network operator. Its shellness varied a lot (increasingly and decreasingly) over the whole time. But it got a balance because the amount of the increase changes and the decrease changes are similar.

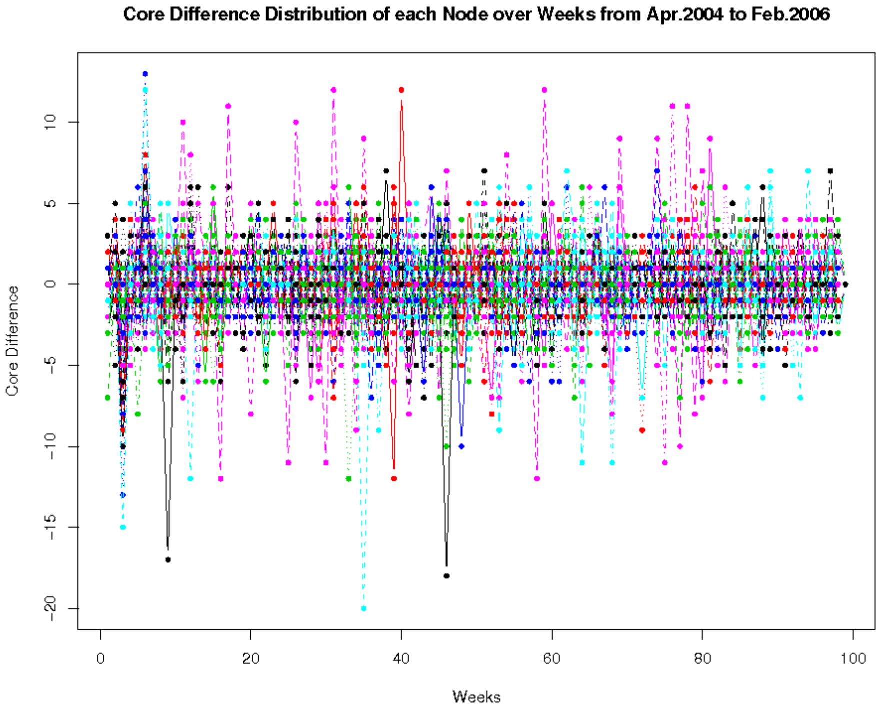


Figure 5.19: Shellness changes of every node in the real Internet network topology over time

Figure 5.20 is an illustration of the amount of the nodes which have the same shell changes at each time step. The x coordinate is the time step from one week to the next week and the y coordinate denotes, how many nodes have the same shell changes. Each line in this figure is colored to represent the shell changes. For instance, the pink line on the top represents no shellness change and the green and black one below it represents that the shellness increased by one and decreased by one. As we can see, this figure is similar to Fig. 5.16. Over half of the nodes have no shell change and most of nodes have small shell changes over time. There are also line pairs in this figure as in Fig. 5.16. For instance, the green and the black line, the gray and blue line etc. Each line pair has approximate values of the quantity of nodes, whose shellness increase and decrease. The percentage of nodes of the relative changes at each time step is very similar. These are also factors that keep the stability of the Internet topology.

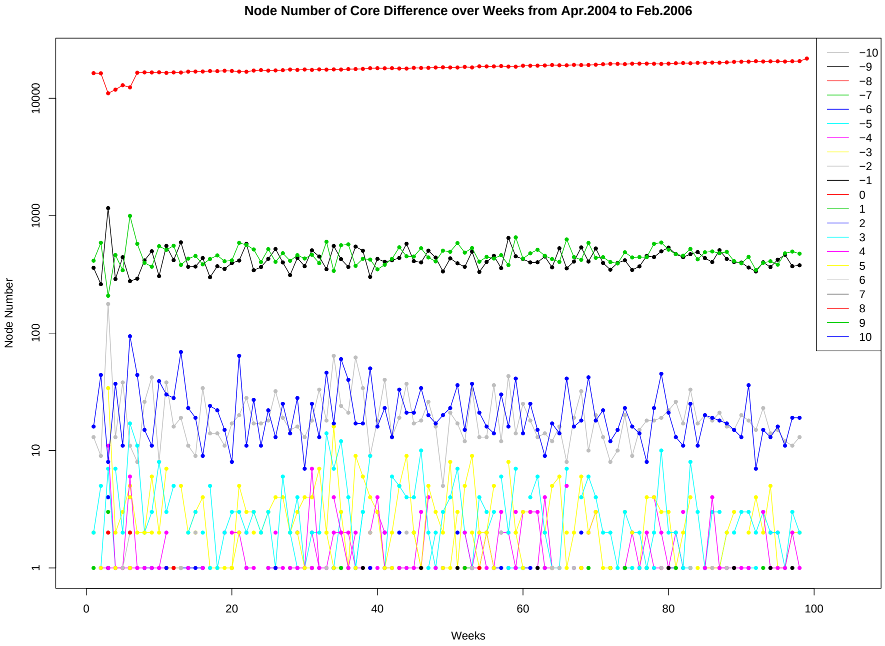


Figure 5.20: The amount of nodes with same shellness changes

The following three figures tell us how nodes move among all shells over time. Fig. 5.21 is for all nodes, Fig. 5.22 is for nodes over 17,000 in 5.21 and Fig. 5.23 is for nodes whose shellness is bigger than 7. The x coordinate denotes time stamps. The figure is composed of 100 pillars. There is a pillar at each time stamp. A pillar is filled with different color of short lines. Each line represents a node. The colors of the line illustrate different shells. The nodes in the first pillar are ordered increasingly by shells and illustrated by different colors. For example, black lines stand for the shell 1 and dark gray ones stand for the shell 2 and so on. The rest 99 pillars are a little different than the first one. The colors of these pillars denote in which shell a node at the first time point was. And a node's current shell is illustrated by the shell tiers. A shell tier in our figures ends when the next shell, which the color represents, in the pillar is smaller than the current shell represented by the color. For instance, in the 100th pillar, the shell 1 starts with the light gray color (new nodes) and ends before the next light gray color; the shell 2 starts with the second light gray color and ends before the third light gray color and so on. The shellness of every node of these pillars at a time point is firstly ordered increasingly by its shellness and is then ordered by its original shellness at the first time point. For example, if a node was in the shell 1 at the first time point and is now in the shell 2 at the 100th time point, then it is in the second black area in Fig. 5.21.

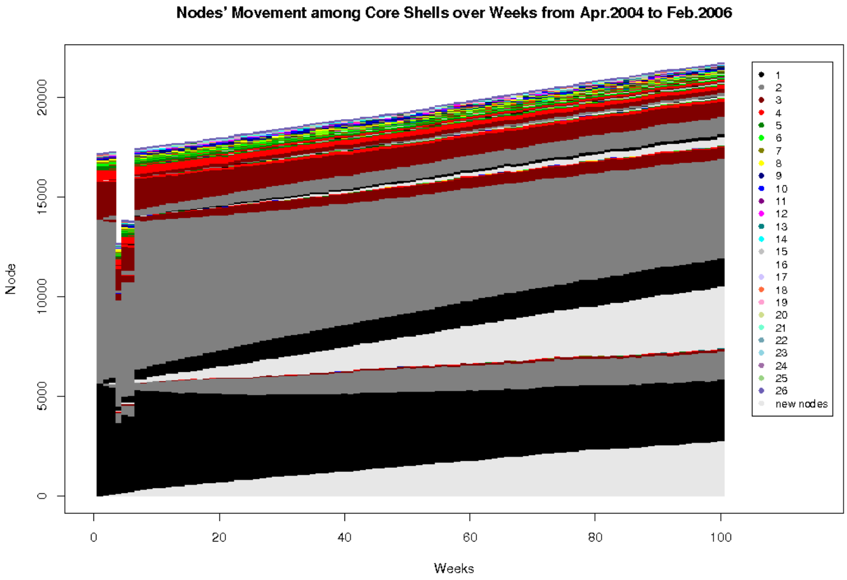


Figure 5.21: Nodes' movement among all shells

To sum up, Fig. 5.21 and Fig. 5.22 tell us how all nodes in the related shells at the start point move to other shells as the time goes by. In Fig. 5.21 we can see, about 75 % of the nodes starting in shell 1 stay in shell 1 over time. Most of the rest move into the shell 2 and only just few nodes move to a higher shell. For the nodes starting with shell 2, 75% of them stay in shell 2 . About half of the rest move into the shell 3 and half fall to the shell 1. Just very few nodes move to a higher shell. The behavior of the nodes in shell 3 is almost the same as the nodes in shell 2. For nodes which are originally in shell 4, about half of them stay in shell 4, 25% of them go one shell up, 25% of nodes fall one shell down and a few nodes go even higher (such as into shell 5, 6) or lower (such as into shell 2). The movement of the nodes which was firstly in shell 5, 6, 7 or 8 is similar (see Fig. 5.22). Over 60% of nodes move 1 or 2 shells up. Most of the rest nodes stay in the original shell and the others fall 1 or 2 shells down.

Fig. 5.23 illustrates another phenomenon. The star marks in the figure are the shell boundaries. As we can see, the maximal shell over time is mainly composed of nodes which were originally in the maximal shell at the first time stamp. The nodes, which were in the shells from 25 to 16 and now are in the maximal shell, are a tiny part of the maximal shell. Of course, our figure is not a complete analysis, there could be an original new node becoming a node in the maximal shell as time goes by. But if a new node would finally move into maximal shell or a very high shells over time, it is also decided by its feature. If the AS is a backbone AS, many other nodes would connect to it as the time goes by and eventually it would become a node in the maximal shell. Furthermore, the number of nodes in these high shells doesn't change much (always about 450 nodes), most changes are in the lower shells, such as shell 1, 2 etc, because of the addition of the new nodes. Therefore, we can say, generally, if a new node is added into the

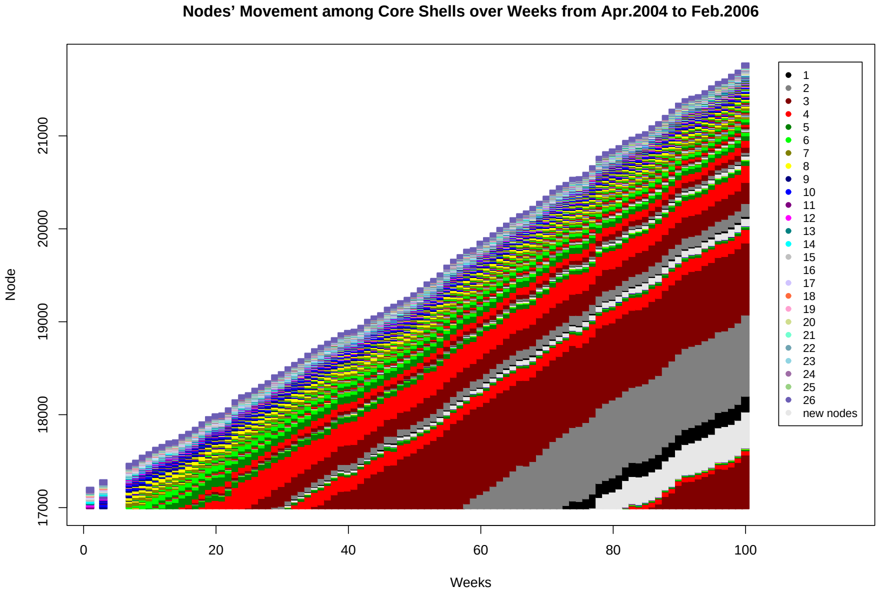


Figure 5.22: Nodes's movement among all shells(above 17000 nodes)

Internet, it is added into low shell area (shell 1, 2 etc.).

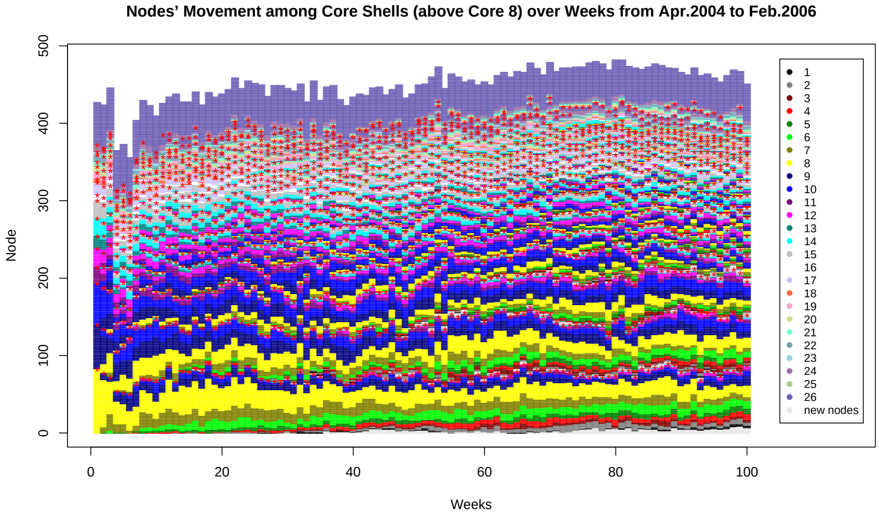


Figure 5.23: Nodes' movement among all shells(above 8th shell)

Besides above mentioned points, we can also find some other phenomena. First of all, in this about 2 years observation, the number of nodes in the Internet at the AS level is a linear growth (except for the jitter in May. 2004). The growth of the quantity of the nodes in each shell is also quasi linear. Secondly, the nodes in the shell 2 and 1 are the majority of the whole Internet (over 70% of all nodes) and there are more nodes in the shell 2 than in the shell 1.

Figure 5.24 illustrates the degree and shell relationship of the Internet over time. The x coordinate denotes the time stamps. The y coordinate is a node's degree ordered by its reversed rank. Every node has an identical rank as described in 3.4.1. A node which has the highest degree has rank 1. In this figure, for a better illustration, we reverse the rank. It means that the node with the smaller degree has smaller rank. The different colors here represent different shells. The white lines in the figure are the degree boundaries. For example, the area from the bottom to the first white line is for nodes with degree 1 and between the first and second line is for nodes with degree 2 and so on. As we can see, as the degree grows, nodes are distributed in a new higher shell, meanwhile they are also distributed in old lower shells. For example, most nodes with degree 2 are distributed in shell 2. A few nodes with degree 2 are distributed in shell 1, these are the nodes which connect to the nodes with degree 1. There are 75% nodes with degree 3 are distributed in shell 3, but also 25% nodes are distributed in shell 2. A part of nodes with degree 4 are distributed in a new shell, shell 4, but the rest of them are also distributed in shells 2, 3. What is also interesting is that as the time goes by, despite a lot of new nodes with different degrees having been added into the Internet, the distribution proportion for different degrees in a shell always stays the same. It is a factor assuring the stability of the Internet topology. Moreover, in this figure we can also see that (as described in Fig. 5.21, 5.22) the increment of the amount of the nodes is linear.

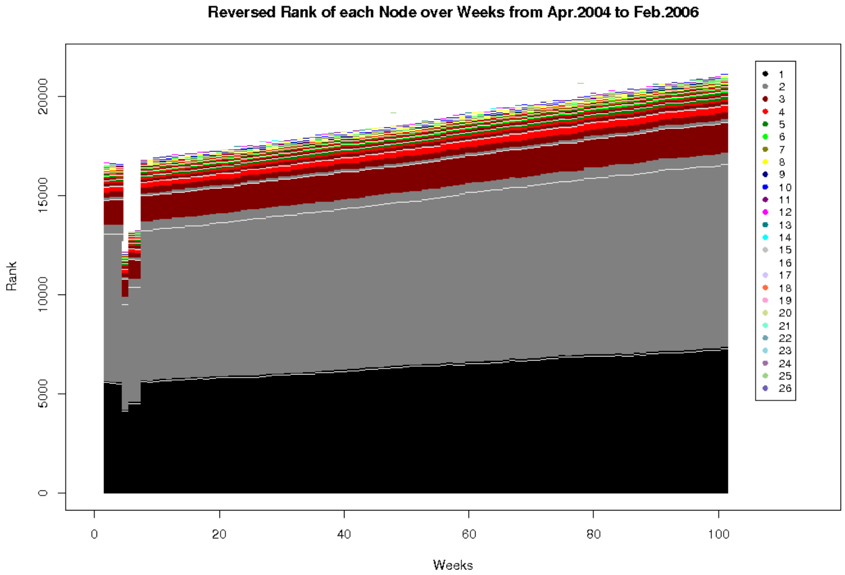
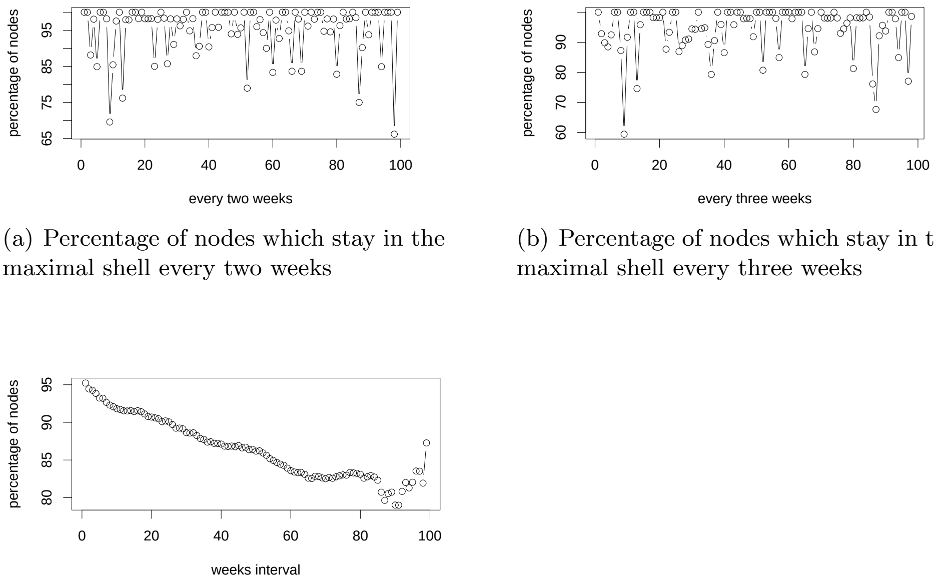


Figure 5.24: Distribution of nodes with different degrees in shells

Figure 5.25 are some figures about the percentage of the nodes, which always stay in the maximal shell as time goes by. The x coordinate is the time interval (every seven days a time point). For example, in Fig.(a) the time interval is one (first week to second week, second to third and so on) and in Fig.(b) the interval is two (first week to third week, second to fourth and so on). Likewise, we can get a series figures with different time intervals. These figures actually show us the overlap of nodes, which always stay in the maximal core. For instance, in Fig.(a) in about half of the weeks, all nodes which were originally in the maximal shell stayed in the maximal shell. In Fig.(b) over the half of the time, the nodes which were originally in the maximal shell stayed in the maximal shell. Fig.(c) is the average of all the figures. The x coordinate denotes the general week interval. '1' means the interval is 'every two weeks' and '2' means the interval is 'every three weeks'. The y coordinate indicates the percentage of nodes, which stay in the maximal shell. For instance, after 4 weeks (start month is random), 94% nodes still stay in the maximal shell. As we can see, over the whole time, there are still 87% nodes which were originally in the maximal shell stayed in the maximal shell.



(a) Percentage of nodes which stay in the maximal shell every two weeks

(b) Percentage of nodes which stay in the maximal shell every three weeks

(c) Average percentage of nodes which stay in the maximal shell over the whole time

Figure 5.25: Percentage of nodes which stay in the maximal shell over time

5.2.4 Rich Club Connectivity-Time

Table 5.2 lists the rich club connectivity of the node whose rank is within 1% of all nodes. This metric tells us the connection inside a small-world group. It is the ratio of the actual amount of connections among nodes and the quantity of links if these nodes are completely connected. The higher this value is, the better these nodes are connected. We chose 1% of the nodes which have the highest degree and calculated the rich club connectivity. We can firstly find one interesting phenomenon: unlike the *dm* p-r-p network, though these 1% nodes have the highest degree, the connections between the nodes in this group is not much, since over the whole time the value of the rich club connectivity is only about 13%. Secondly, from the start (Apr.1 2004) of our observation data of the Internet topology to the end of that year, the value of the rich club connectivity stayed at $13-14\%$ (except for the jitter in May 2004). It was about 12% in the whole year 2005. From

Rich club connectivity $\phi(r = 1\%)$					
1-5 week	0.148407	0.146236	0.150488	0.215674	0.186742
6-10 week	0.188823	0.138442	0.142857	0.142270	0.137969
11-15 week	0.138450	0.138704	0.141116	0.140481	0.140606
15-20 week	0.143243	0.138610	0.140596	0.139718	0.139840
21-25 week	0.139779	0.144739	0.142857	0.144238	0.145860
26-30 week	0.147660	0.147066	0.145476	0.145535	0.144319
31-35 week	0.142226	0.140935	0.140417	0.140118	0.133462
36-40 week	0.132444	0.137060	0.136046	0.130827	0.134670
41-45 week	0.130047	0.129953	0.125930	0.125873	0.125545
46-50 week	0.126889	0.127915	0.122895	0.123925	0.123711
51-55 week	0.123500	0.127518	0.127943	0.128938	0.129493
56-60 week	0.127057	0.125212	0.126542	0.125171	0.125377
61-65 week	0.122010	0.124229	0.123781	0.123590	0.122851
66-70 week	0.120324	0.122860	0.122128	0.119917	0.121234
71-75 week	0.119560	0.120009	0.120530	0.120163	0.119788
76-80 week	0.121899	0.122166	0.119019	0.120399	0.120939
81-85 week	0.120392	0.118077	0.117942	0.117626	0.117232
86-90 week	0.116829	0.116175	0.115688	0.116362	0.115540
91-95 week	0.114801	0.114540	0.112360	0.114040	0.112963
96-100 week	0.112007	0.111239	0.111571	0.111402	0.109170

Table 5.2: Rich club connectivity against rank r of the Internet

then to the end (Feb. 2006) of our observation data it decreased by 1 too (11%). We assume here that the value of the rich club connectivity (against the rank of 1%) reduces by 1% every year. In the future, we can obtain more data of the Internet topology to prove this assumption.

5.2.5 Time Series Statistics

Over the whole time, there are on the average (dynamic average $\overline{a}$) 19510 nodes in the Internet. From the start time to the end of the observation the geometric average development rate is 100.24%. The average amount of nodes in the shell 1, 2 and the maximal shell are 6545, 9233 and 25 and the respective geometric average development rates are 100.27%, 100.22% and 99.83%. The average highest shell is 25 and the geometric average development rate for the highest shell during the observation is 99.96%. There are average 17520 nodes in the low degree ($\leqslant 5$) group and the geometric average development rate is 100.24%, meanwhile the respective values for the high degree ($\geqslant 100$) group are 67 and 100.19% (see the complete statistics in the Appendix A-1). Through the statistic we can see the total number of nodes, the amount of nodes in each shell, for each degree group increased slightly (about 0.02%). The amount of shells doesn't increase during the observation.

5.2.6 Summary

In this section, our goal is to find out how the Internet evolves over time. For example, as the new nodes are added into the network and connections are increasing, how the degree of each node changes? How the original nodes in the network move to other shells as time goes by? How the new nodes are added to different shells? Therefore, we observed and analyzed the Internet at the AS level from Apr. 1, 2004 to Feb. 28, 2006 with the metrics we chose: degree, frequency, shellness and rich club connectivity.

In our analysis, we found that the Internet topology has a steady structure. All the degree distribution are similar over the whole time. Nodes with low degree (degree 1 and 2) constitutes the majority of the network (5.18). Over half of nodes have no degree changes (5.17). Most of the rest have small degree changes and these changes took place in the low degree groups (5.16, 5.18). Certainly there are nodes whose degree changes a lot. These are the nodes with high degree and most of them belong to the backbone ASes. If the degree of a node changes a lot at a time point, it will adjust its degree the next few steps (5.15). The changes of the quantity of the nodes which has the same degree changes is quasi symmetrical the whole time, which means if the degrees of some nodes increase, there is definitely an approximately equal quantity of nodes, whose degree decrease (5.17). This is the reason that the network topology doesn't change much. Compared with the first time point, there are altogether 6472 new nodes added into the network and 97% of them are with low degree (< 5). Therefore, the degrees of most nodes increase slightly. Besides, the number of the nodes with high degree (backbone ASes) stay quasi the same over the whole time (5.18).

The nodes in shell 2 and the shell 1 are the majority of the whole Internet (over 70% of the nodes) and there are more nodes in shell 2 than in shell 1 (5.21). Most (over half) of nodes have small shellness changes. The nodes which have bigger shellness changes are the ASes, such as operators, in the backbone of the network (5.19). The changes of the amount of the nodes for each shell is also quasi symmetrical over the whole time (5.20). It assure that the shell topology of the Internet doesn't change much as time goes by. During the observation time, over half of the nodes stayed in the original shell and the rest of them moved to the shell near their original shell (they go $1-2$ shells up or down)(5.21, 5.22, 5.23). There are few nodes which moved dramatically among shells. The majority of the maximal shell is composed of the nodes (87% of them) which are originally in the maximal shell at the first time point. The small rest part is composed of the nodes moving from shell 25-16 over time (5.23). If a new node (compared with the first time point) is added into the network, it participates firstly in the low shell. As time goes by, it will stay in its shell or move to a higher shell according to its feature of its own, for instance, if it is a backbone AS, it starts with low shell and will eventually move into a very high shell (5.23). The quantity of the nodes in the Internet at the AS level is a linear growth (except the jitter in May. 2004). The growth of the amount of the nodes in each shell is also quasi linear (5.21).

We also found some relationship between the degree of a node and the shellness of it. Over the whole observation, a part of the nodes with degree k are distributed in the shell k, meanwhile the other part of the nodes are distributed in shells $k-1$, $k-2$, $\cdots$, 1. As time goes by, despite a lot of new nodes with different degrees were added into the Internet, the distribution proportion for different degrees in a shell always stayed the same (5.24).

Besides above mentioned points, we also found that the value of the rich club connectivity decreased by 1 every year from the year 2004 to the year 2006 (see table 5.2). We assume that this value decreases by 1 every year until it get some kind of balance. However we need more observation data for this assumption in the future.

Chapter 6

Proposal of an Evolving Model for the Internet at the AS Level

In 5.2 we found some regularities about how the nodes move among the shells over time. During the observation time, over half of the nodes stayed in the original shell and the rest of them moved to a shell near their original shell (they go $1-2$ shells up or down). There are few nodes which moved dramatically among shells. The majority of the maximal shell is composed of the nodes (87% of them) which are originally in the maximal shell at the first time point. The small rest part is composed of the nodes moving from shell 25-16 over time. If a new node (compared with the first time point) is added into the network, it participates firstly in the low shell. As time goes by, it will stay in its shell or move to a higher shell according to its feature of its own. The quantity of the nodes in the Internet at the AS level is quasi a linear growth (except for the jitter in May. 2004). The growth of the amount of the nodes in each shell is also quasi linear. In this chapter we are trying to model some of regulations so that we can propose an evolving model for the Internet at the AS level.

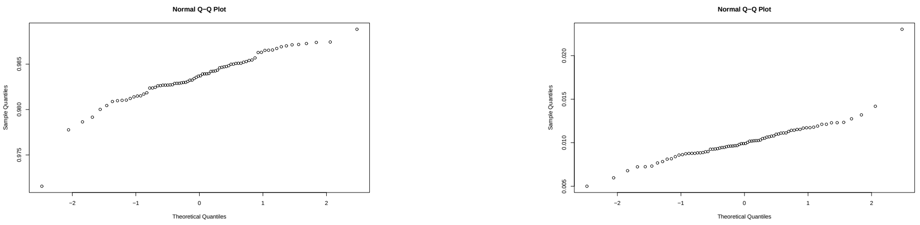
6.1 I.Attempt

In our first attempt we came up with the idea of the following matrix that describes the above mentioned phenomena and it is referred to as the transformation matrix M. There is such matrix for each time step (from one time point to the next time point). The matrix is composed of 27 rows and 27 columns. Each entry in the matrix denotes how many nodes there are moving from one shell to another shell for a time step. For example, entry $x_{2,3}$

denotes the amount of the nodes moving from the shell 2 to the shell 3 and entry $x_{3,0}$ denotes the quantity of the nodes leaving the Internet (from the shell 3 to the shell 0). Since over half of the nodes stay in the original shell and the rest move near their original shells, we focus on the $x_{i,i}$, $x_{i+1,i}$, $x_{i+2,i}$, $x_{i,i+1}$, $x_{i,i+2}$ entries near the diagonal. Other entries are assigned 0. The sum of a row $x_{i,n}$, $(n = 0 \dots 26)$, denotes the number of nodes moving from the shell i to other shells and the sum of a column $x_{n,i}$ denotes the number of nodes moving from other shells to the shell i. Therefore, for a time step the change of the quantity of nodes in the shell i is $\sum_{n=0}^{26} x_{i,n} - \sum_{n=0}^{26} x_{n,i}$. With the value of the quantity of nodes in each shell at the first time point, we can calculate how many nodes there will be in each shell at the second time point.

$$\mathbf{M} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & 0 & 0 & \dots \\ x_{10} & x_{11} & x_{12} & x_{13} & 0 & \dots \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & \dots \\ 0 & x_{31} & x_{32} & x_{33} & x_{34} & \ddots \\ 0 & 0 & x_{42} & x_{43} & x_{44} & \ddots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}$$

As we can see, if we can predict the value of the matrix, M_t,at time point t , then we can calculate the changes of number of nodes in each shell at each time point, adding them all to the number of nodes at the start point, we can tell how is the Internet constructed at time point t.



(a) Residual quantile quantile plot for entry x_{11}

(b) Residual quantile quantile plot for entry x_{12}

Figure 6.1: Residual q-qplots for entry in M matrix

We observed the Internet topology every seven days from Apr. 2004 to Feb. 2006. All together we have 100 time points and 99 time steps. For each time step, we calculated the corresponding matrix in order to see how every entry in the matrix changes over time. In our first attempt we tried to obtain a linear model by curve fitting for each entry in the matrix over time, so that we can predict the value of this entry. Fig. 6.1 is the residual quantile-quantile plot for the linear model for entry $x_{1,1}$ and $x_{1,2}$. A q-qplot is a plot of the quantiles of the first data set against the quantiles of the second data set. By a quantile, we mean the fraction (or percent) of points below the given value. That is, the 0.2 (or 20%) quantile is the point at which 20% percent of the data fall below and 80% fall above that value. The q-qplot is used to test if the two data sets come from populations with common distribution. As described in 3.5, if the nodes in the q-qplot falls close to a straight line, the residuals are normal. Likewise, we calculated a linear model for each entry in matrix M. Now we can calculate and predict the matrix M at any time point. For instance, if we need to simulate the Internet at the 101 time point starting from Apr. 2004, what we need to do is as follows:

1. Calculating the corresponding matrix $M_1, M_2, \cdots, M_{99}$ for each time step
2. Calculating the matrix M_{100} according to the linear models
3. Calculating the changes of the amount of nodes in each shell for each time step by $\sum_{n=0}^{26} x_{i,n} - \sum_{n=0}^{26} x_{n,i}$, $(i = 0 \cdots 26)$ in $M_{100}$and then add them together. It is then the whole changes after 100 time steps.
4. Adding the changes to each shell of the Internet at the start point, we

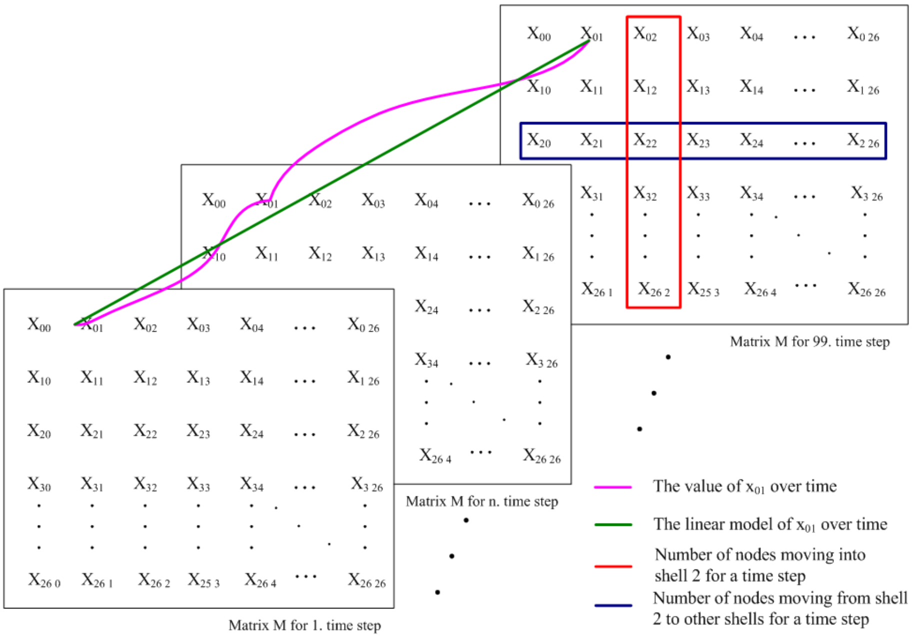

X00 X01 X02 X03 X04 ... X0 26
X10 X11 X12 X13 X14 ... X1 26
X20 X21 X22 X23 X24 ... X2 26
X30 X31 X32 X33 X34 ... X3 26
X26 0 X26 1 X26 2 X25 3 X26 4 ... X26 26
Matrix M for 1. time step
Matrix M for n. time step
Matrix M for 99. time step
The value of x01 over time
The linear model of x01 over time
Number of nodes moving into shell 2 for a time step
Number of nodes moving from shell 2 to other shells for a time step

Figure 6.2: Illustration of the matrices

simulate the Internet at the 101 time point.

The simulated node distribution in each shell is very similar to the real Internet topology at the AS level for the first 100 time steps. However, we found that after 200 time steps, there are negative values in the high shell, such as shell 26, 24. Because the linear model for each entry in the matrix is in the form of $y = ax + b$. The slope a could be negative, so the predicted value could be negative. The total changes in each shell is the sum of each change in the shell, hence it could also be negative then. Therefore, we concluded that the linear model is not suitable for the transformation matrices and the evolving model.

6.2 II.Attempt

We found in the first attempt that the linear model is not suitable for our evolving model. In the second attempt we used the idea of nodes' change proportions instead of the linear model. We also proposed a change matrix M_p composed of 27 rows and 27 columns in this part. It is similar to the matrix we proposed in the first attempt.

$$\mathbf{M_p} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & \cdots \\ x_{10} & x_{11} & x_{12} & \cdots \\ x_{20} & x_{21} & x_{22} & \cdots \\ x_{30} & x_{31} & x_{32} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

We observed again the nodes' movement for a time step. Fig. 6.3 are some example figures. As we can see, the percentage of nodes moving from one shell to another always varied around an average value over time (e.g. the figure shell 1 to shell 1 or shell 26 to shell 26). Therefore, we came up with the idea of a matrix that describes the average proportion of nodes moving from one shell to another and change the state with the matrix to obtain the topology graph at the next time point.

6.2.1 Transformation Matrix

Firstly, we have a matrix M, where each entry describes how many nodes moving from one shell to another shell at each time point, e.g. x_{12} denotes the amount of nodes moving from shell one to shell two. Every entry in matrix M_p from row 1 to row 26 is calculated on the basis of matrix M:

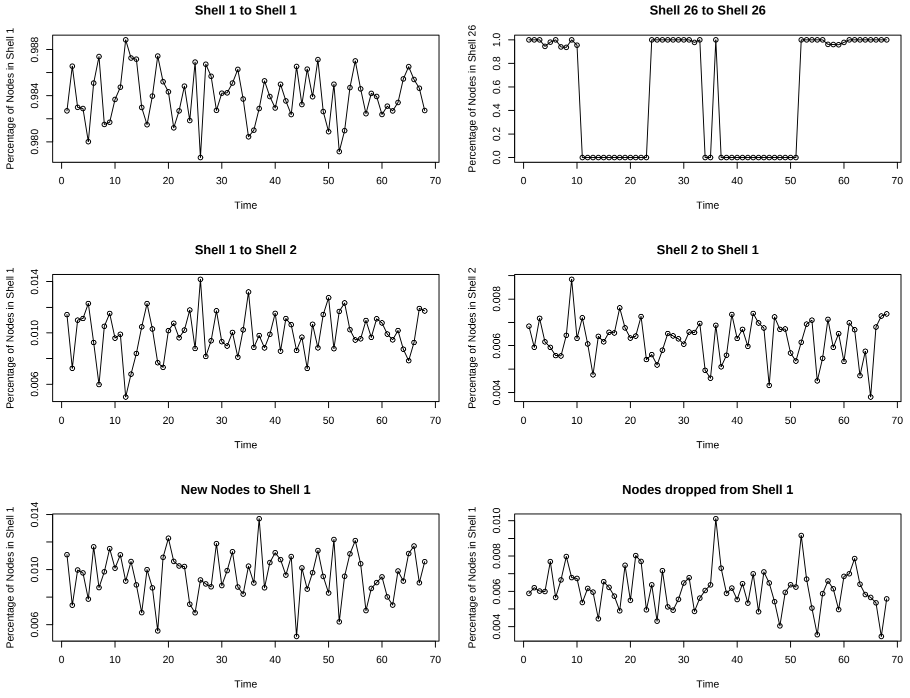


Figure 6.3: Observation of proportion of nodes' movement among all shells for 70 time steps

$$x_{i,j}\ in\ matrix\ M_p = \frac{x_{i,j}\ in\ matrixM}{\sum_{k=0}^{26} x_{i,k}\ in\ matrixM}(i = 1 \cdots 26, j = 0 \cdots 26)$$

The first row in matrix M_p is calculated as follows:

$$x_{0,j}\ in\ matrix\ M_p = \frac{x_{0,j}\ in\ matrixM}{\sum_{k=0}^{26} x_{j,k}\ in\ matrixM}(j = 1 \cdots 26)$$

And

$$x_{0,0}\ in\ matrix\ M_p = 1$$

$\sum_{k=0}^{26} x_{i,k}\ in\ matrix\ M\ (i \neq 0)$ is the amount of nodes moving from shell i to all shells, which is actually the total amount of nodes in shell i. $x_{i,j}\ in\ matrix\ M$ is the amount of nodes moving from shell i to the shell j. Therefore, each entry $x_{i,j}\ in\ matrix\ M_p$ describes the change proportion moving from shell i to shell j of the total changes of nodes moving from shell i to all shells (or total amount of nodes in the shell i) for each time step. For instance, the $\sum_{k=0}^{26} x_{1,k}\ in\ matrix$ is the total changes of node moving from the shell 1 to all shells. The entry $x_{1,2}$ in matrix M_p is the change percentage (nodes moving from the shell 1 to the shell 2) of the total change (nodes moving from the shell 1 to all shells). Furthermore, the first row in the matrix denotes the new nodes' proportion added into each shells. As we can see in the above formula, the new nodes' distribution is decided by the scale of each shell, which means a shell with more nodes is getting more new nodes. And the first column denotes the nodes dropped from each shells.

We observed the Internet topology every seven days from Apr. 2004 to Feb. 2006. All together we have 100 time points and 99 time steps. For each time step, there is a matrix M_p and we found in the matrices each value of $x_{i,j}$, the change proportion, is close over the whole time. Therefore, we decided to calculate the average value of each $x_{i,j}$ over time. And then we obtained an average transformation matrix (from all matrices for each time step). We refer this average matrix as the universal transformation matrix $\overline{M_p}$.

Now we have a function f, that calculates the change of each shell for each time step. For instance, the function f calculates how many nodes moving from the shell i ($i = 0 \cdots 26$) to the shell 2 and then sum them up to get the new state (new amount of nodes) of the shell 2. In order to simulate the Internet at next time point, we only need the start state of the topology and using function f calculate the new state at the next time point with the transformation matrix $\overline{M_p}$. The theoretical runtime for Algorithm 4 is $O(t(s+1)^2) + O(t(s+1)) \subset O(ts^2 + ts)$, where t is the requested time point and s is the maximal shell. The $(s+1)^2$ in the first part is for the calculation of the movement of nodes in function f and the $(s + 1)$ in the second part is for the sum process in function f.

Algorithm 4: To simulate the Internet topology at the AS level at the requested time point

```
Data: Start state of an Internet topology, the time point of the
      simulated Internet topology
Result: Simulated Internet topology at the according time point
for $i \leftarrow 1$ to the request time point do
    $StartState \leftarrow f(StartState, \overline{M_p})$ ; /* function $f$ changes the
    start state to the next start with the transformation
    matrix */
```

6.2.2 Refinement 1

In our analysis, we found that if there is a node that should fall to a lower shell, the node which was in the lower shell has more probabilities to be picked to move to the lower shell. It means that the new state not only depends on the present state but also the previous states. In order to include this feature in our transformation matrix, we average the nodes' state (or nodes' shellness) at every time point as follows: a node's shellness at the present time point is the average value of shellnesses of this node from the present time point to the next 9 time points. We call the interval the *average window size* (see the illustration in Fig 6.6).

Algorithm 5: To obtain the refined universal transformation matrix $\overline{M_p}$

```
Data: Observed nodes' shellness at each time point
Result: The refined universal transformation matrix $\overline{M_p}$
foreach time $t$ from 1 to 91 do
    foreach node $n$ do
        $n_t.shell \leftarrow$ the shell of node $n$ at time point t;
        if $n_t.shell == 0$ then
            $n_t.shell = 0$;
        else
            $n_t.shell = avg(\{n_{t+i}.shell | i = 0 \dots 9,\ n_{t+i}.shell \neq 0\})$ ;
            /* if a node's shellness is not zero, it is
            assigned the average of the next 10 shellness
            including itself */
foreach time step $t$ from 1 to 90 do
    $s$ ; /* $s$ is a matrix as $s[a, b, t]$ with the number of nodes
    in shell $a$ at time point $t$ and in shell $b$ at time point
    $t + 1$ */
    $vs$ ; /* $vs$ ia a matrix as $vs[a, t]$ with the number of nodes
    in shell $a$ at time point $t$ */
    foreach row $r$ of $m[\ ,\ , t]$ do
        if $r == 0$ then
            foreach column $c$ of $m[\ ,\ , t]$ do
                $m[r, c] \leftarrow \frac{s[r,c,t]}{vs[c,t]}$ ; /* calculate the first row of
                transformation matrix described in (6.2.1) */
        else
            foreach column $c$ of $m[\ ,\ , t]$ do
                $m[r, c] \leftarrow \frac{s[r,c,t]}{vs[r,t]}$ ; /* calculate the remaining rows
                of transformation matrix described in (6.2.1)
                */
$\overline{M_p} \leftarrow$ the average matrix of $m$ along dimension $t$
```

The main algorithm of the refined universal transformation matrix $\overline{M_p}$ is list in algorithm 5. In this case, our average window size is 9. The average shellness of a node reflects the node's location preference over time. Indirectly, it reflects the phenomenon we mentioned above. Since the last 9 time points have no more enough data in a time interval to average, we ignored this last 9 time points. Now we have average nodes' states (or shellnesses) at 91 time points. And then we can obtain all M_p matrices for 90 time steps and the universal transformation matrix $\overline{M_p}$ as described in 6.2.1. The theoretical runtime for Algorithm 5 is $O(wnt_1)+O(tn)+O(2t(s+1)^2) \subset O(wnt_1 + nt + s^2t)$, where t_1 is the time point, t is the time step, s is the maximal shell, n is the number of nodes and w is the average window size. The $O(wnt_1)$ part is for averaging every node's shellness with the average window size. The $O(tn)$ part is for lines 11 and 12 in the algorithm to store the nodes' movement for a time step. And the $O(2t(s+1)^2)$ is for the lines from 13 to 20 in the algorithm, which calculate the transformation matrix.

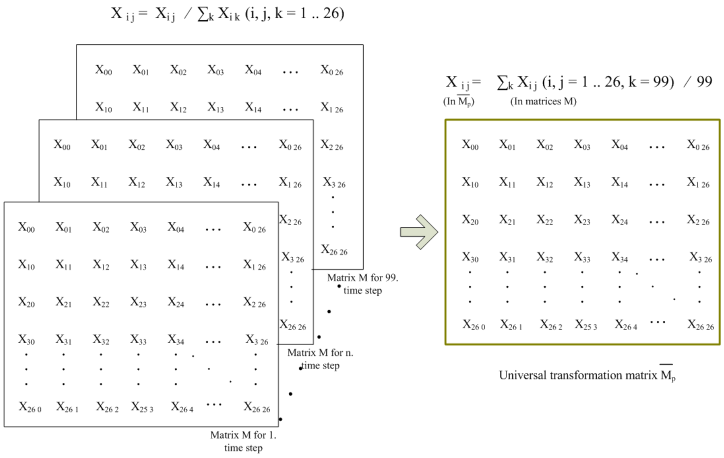

X ij = Xij / ∑k Xik (i, j, k = 1 .. 26)
X ij = ∑k Xij (i, j = 1 .. 26, k = 99) / 99
(In $\overline{M}_p$)
(In matrices M)
X00 X01 X02 X03 X04 … X0 26
X10 X11 X12 X13 X14 … X1 26
X20 X21 X22 X23 X24 … X2 26
X30 X31 X32 X33 X34 … X3 26
X26 0 X26 1 X26 2 X25 3 X26 4 … X26 26
Matrix M for 99. time step
Matrix M for n. time step
Matrix M for 1. time step
Universal transformation matrix $\overline{M}_p$

Figure 6.4: Illustration of the matrices

The following is the refined average universal transformation matrix.

$$\overline{\mathrm{M_p}} = \left(\begin{array}{ccccccccccccccccccccccccccc}
1 & 0.009749376 & 0.003884519 & 0.0007412073 & 0.0005398028 & 0.0004836206 & 0.0005042503 & 0.0002892194 & 0 & 0.0001604621 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.006257029 & 0.983613 & 0.01012991 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.001299952 & 0.006286487 & 0.9881892 & 0.004224392 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0003779915 & 0 & 0.01519260 & 0.9765766 & 0.0078528 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0002232605 & 6.116856e-05 & 0 & 0.0202409 & 0.9661194 & 0.01335527 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0005222616 & 0 & 5.370569e-05 & 0.0001048710 & 0.02552449 & 0.9511097 & 0.02268498 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0003780343 & 0 & 0 & 0 & 0.0001087429 & 0.03022045 & 0.9387932 & 0.03031564 & 0.0001839088 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0003057524 & 0 & 0 & 0 & 0 & 0.0002910043 & 0.03727445 & 0.9195615 & 0.04256725 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0001495215 & 0 & 0 & 0 & 0 & 0 & 0.0003157139 & 0.04116937 & 0.9190571 & 0.03930829 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0004020468 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0004741206 & 0.04650225 & 0.9077325 & 0.04488907 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0005215649 & 0.04831111 & 0.9140272 & 0.03689184 & 0.0002482622 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00068604 & 0.06512451 & 0.8828113 & 0.05137811 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001649775 & 0.04983983 & 0.8899495 & 0.05856086 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0006578947 & 0.06620896 & 0.8743885 & 0.05874468 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.003124013 & 0.06103012 & 0.8962785 & 0.03956732 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001644737 & 0.07080718 & 0.8522568 & 0.0752913 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001096491 & 0.05467091 & 0.885583 & 0.05864968 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001196172 & 0.0009398496 & 0.06194619 & 0.8651431 & 0.07077464 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001754386 & 0.07310004 & 0.8590798 & 0.06606577 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002392344 & 0.06342342 & 0.8476299 & 0.08655434 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001196172 & 0.05330026 & 0.8641955 & 0.08130802 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002804359 & 0.09000031 & 0.8549555 & 0.05223986 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.07122348 & 0.8418308 & 0.08694573 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002192982 & 0.001461988 & 0.07392962 & 0.8324625 & 0.08995294 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002192982 & 0.001315789 & 0.05141638 & 0.899313 & 0.04576186 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02356361 & 0.9007623 & 0.07567408 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0004 & 0.00938 & 0.9058
\end{array}\right)$$

6.2.3 The Simulator and Evaluation of the Proposed Evolving Model

Algorithm 6: Internet topology simulator

```
   Data: Start state of an Internet topology, the time point of the
         simulated Internet topology
   Result: Simulated Internet topology at the according time point
1  $vsi \leftarrow$ the node number in each shell at the start time point;
2  $\overline{m} \leftarrow$ the universal transformation matrix;
3  $round \leftarrow$ the time point of the simulated Internet topology;
4  $vs \leftarrow vsi$;
5  foreach round t do
6      foreach row r in $\overline{m}$ do
7          if $r == 0$ then
8              foreach column c in $\overline{m}$ do
9                  $s[r,c] \leftarrow \overline{m}[r,c] \times vs[c]$ ; /* calculate the number
                   of new added nodes in each shell at a time
                   point */
10         else
11             foreach column c in $\overline{m}$ do
12                 $s[r,c] \leftarrow \overline{m}[r,c] \times vs[r]$ ; /* calculate the number
                   of nodes moving from shell r to shell c at a
                   time point */
13     foreach element e in vs do
14         $s[e, \,]$ ; /* e-th row in $s$ */
15         $s[\, ,e]$ ; /* e-th column in $s$ */
16         $vs[e] \leftarrow vs[e] - sum(s[e, \,]) + sum(s[\, ,e])$ ; /* calculate the
           new state of nodes as described in (6.2.1) */
```

In order to evaluate our evolving model for the Internet at the AS level, we programmed a simulator to accomplish the task of simulating the Internet topology. The simulator is actually a realization of the proposed evolving model. We have a start state of the Internet topology, including the number of nodes and how these nodes are distributed in each shells. And then we change these states with our universal transformation matrix $\overline{M_p}$ to get the state of the topology at the next time point. Executing this repeatedly we can obtain the simulated topology at any time point. The main idea of the simulator is listed as follows. The theoretical runtime for Algorithm 6 is $O(t(s+1)^2) + O(2st(s+1)) \subset O(s^2t)$, where t is the time point and s is the maximal shell. The $(s+1)^2$ is for lines from 7 to 12 in the algorithm to calculate a matrix of the change of the number of nodes from one shell moving to another shell. The $O(2st(s+1))$ is for lines from 13 to 16 in the algorithm, which calculate the changes of the amount of nodes in each shell. To calculate the number of nodes in each shell at the requested time point, it costs the CPU, for example, 0.38 seconds for 50 rounds in the algorithm or 2.2 seconds for 500 rounds. To simulate each node's shellness at the requested time point, it costs the CPU, for instance, 2.6 secondes for 50 rounds or 98.7 seconds for 500 rounds.

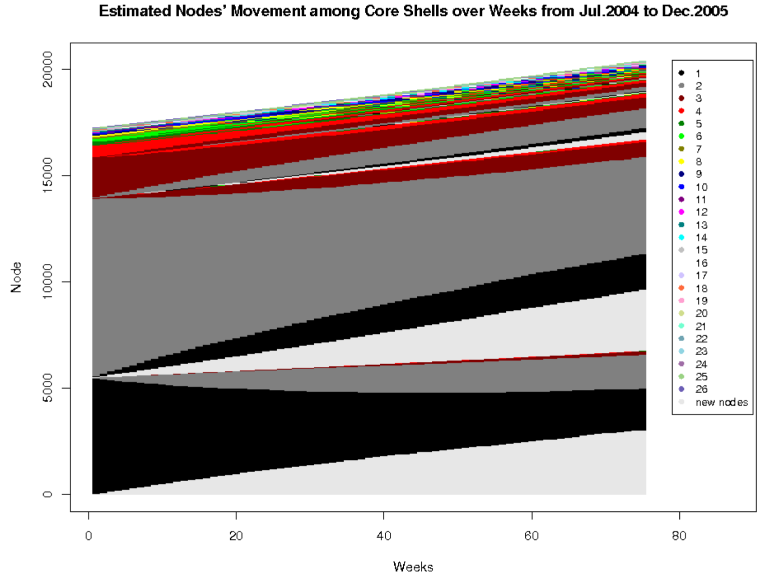


Figure 6.5: simulated nodes' movement among all shells for 90 time steps

Figure 6.5 is the node's movement among shells of simulated Internet topology starting from the original 15th time point, because there are some jitters in the first 14 time points. The x coordinate denotes time stamps. The figure is composed of 90 pillars. There is a pillar at each time stamp. A pillar is filled with different color of short lines. Each line represents a node. The colors of the line illustrate different shells. The nodes in the first pillar are ordered increasingly by shells and illustrated by different colors. For example, black lines stand for shell 1 and dark gray ones stand for shell 2 and so on. The rest pillars are a little different than the first one. The colors of these pillars denote in which shell a node at the first time point was. And a node's current shell is illustrated by the shell tiers. A *shell tier* in our figures ends when the next shell, which the color represents, in the pillar is smaller than the current shell represented by the color. For instance, in the 75th pillar, the shell 1 starts with the light gray color (new nodes) and ends before the next light gray color; the shell 2 starts with the second light gray color and ends before the third light gray color and so on. The shellness of every node of these pillars at a time point is firstly ordered increasingly by its shellness and is then ordered by its original shellness at the first time point. For example, if a node was in the shell 1 at the first time point and is now in the shell 2 at the 75th time point, then it is in the second black area in Fig. 6.5. As we can see, it is very similar to the nodes' movement among shells of the real Internet in 5.21. The proportions of nodes with the original shellness moving into other shells are almost the same as the real Internet illustrated in 5.21.

As the result of the simulated Internet topology shows, this model for the Internet at the AS level is successful. It can precisely simulate nodes' movement among shells over time and simulate the Internet at the AS level in the future, e.g. after 150 time steps or any other time steps. But we should also notice that there are some residuals in the simulated topology graphs using our simulator. Because in the simulating process, the nodes' distribution by multiplying the universal transformation matrix could be not integral and we have a function in our simulator to make them integral somehow, which leads to deviations.

6.2.4 Refinement 2

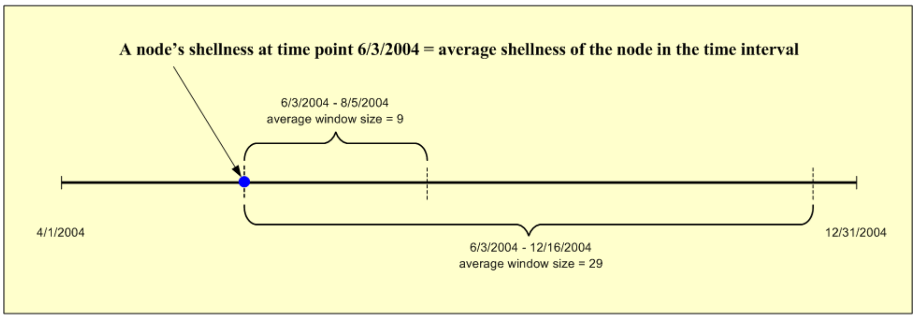


Figure 6.6: Illustration of average window size

In our experiments for the evaluation of our simulated Internet topology compared with original one, we found the *average window size* (6.2.2) used for calculating the universal transformation matrix $\overline{M_p}$ also influences the precision of the simulated topology. We found that the bigger the average window size is, the more precise the simulated topology is (see Fig 6.6). Figure 6.7 shows us in each shell how many percentages of nodes, compared with the state at the start time point, stay at their own shell over time. For example, in the real Internet and the normalized Internet at time point 20,

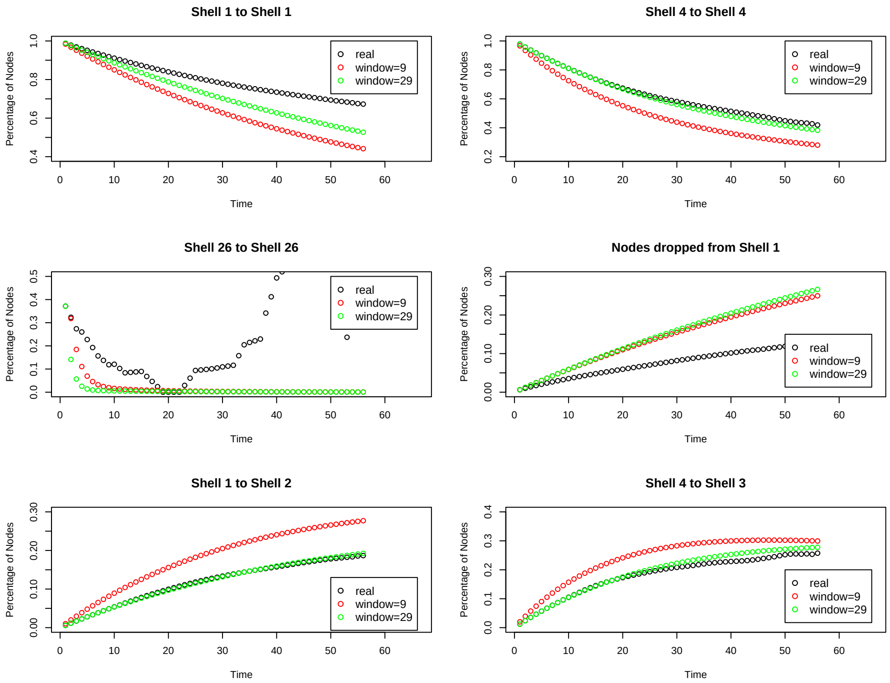


Figure 6.7: Comparison between the real Internet topology and the simulated topology with different average window size

80% nodes, which were in shell 1 at the first time point, are now staying at shell 1 and 87% nodes, which were in shell 2 and now are staying at shell 2. Different colors represent the real Internet topology at the AS level and the simulated topology with different average window size. As we can see, for instance, with bigger average window size, the simulation of the percentage of nodes moving from the shell 4 to the shell 4 for a time step is more close (more precise) to the real Internet topology. And it is also true for nodes in all other shells. Because the bigger the average window size is, the more representation the average value of the shell has, where a node prefers to stay at. Through experiments, we found that if we set the average window size to 29, the simulated topology graph is pretty precise.

6.2.5 Refinement 3

As we can see in Fig 6.7, after our two refinements, the simulated topology graph is more closer to the real Internet topology. The essence of our refinements is to reflect the nodes' feature of the location preference over time, namely, a node has higher probabilities to be picked to move back to its original shell. The two refinements improved the proposed algorithm. However, there are still deviations by picking out the nodes moving from one shell to another shell. In our algorithm, we simulate the Internet topology at a certain time point by repeatedly calculating the new state of nodes (starting from the start state) for the next time point with the universal transformation matrix $\overline{M_p}$ until the requested time point. In every calculation of the new state, there will be deviations in picking out the nodes. Through the repetition, the deviation is getting bigger. In order to address this problem, we came up with the third refinement. We illustrated our idea in Fig. 6.8.

Previously, we calculated the average universal transformation matrix for each step $\overline{M_p}$, we call it now as **the average universal transformation matrix with step** 1 or $\overline{M_p^1}$. Meanwhile, we calculate 6 different transformation for different steps. For example, we have the nodes state in each shell at the first time point and at the 51th time point. Then we can calculate a transformation matrix with step 50. Likewise, we can calculate a transformation matrix with step 50 using the nodes state at the second point and at the 52th time point, $\cdots$, at the 50th time point and at the 100th time point. By averaging these 50 transformation matrices with step 50, we obtain an average universal transformation matrix $\overline{M_p^{50}}$. Likewise, we can obtain matrices $\overline{M_p^{25}}$, $\overline{M_p^{12}}$, $\overline{M_p^{6}}$, $\overline{M_p^{3}}$. Together with the $\overline{M_p^{1}}$, we can calculate or predict the Internet topology at any time point. For instance, we would like to simulate the Internet topology at the 366th time point starting from the observation start. We repeat algorithm 6.4 seven

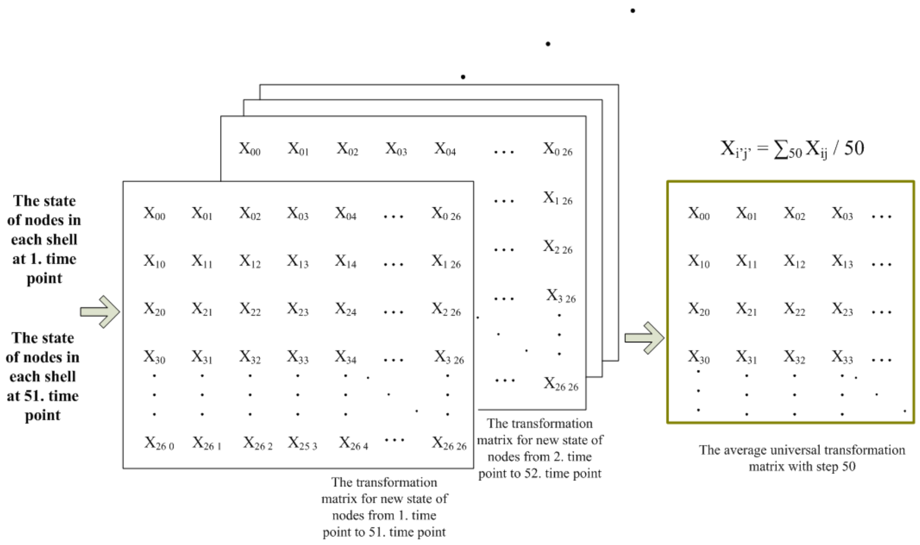

The state of nodes in each shell at 1. time point
The state of nodes in each shell at 51. time point
X00 X01 X02 X03 X04 ... X0 26
X10 X11 X12 X13 X14 ... X1 26
X20 X21 X22 X23 X24 ... X2 26
X30 X31 X32 X33 X34 ... X3 26
X26 0 X26 1 X26 2 X25 3 X26 4 ... X26 26
The transformation matrix for new state of nodes from 1. time point to 51. time point
The transformation matrix for new state of nodes from 2. time point to 52. time point
X26 26
Xi'j' = ∑50 Xij / 50
X00 X01 X02 X03 ...
X10 X11 X12 X13 ...
X20 X21 X22 X23 ...
X30 X31 X32 X33 ...
The average universal transformation matrix with step 50

Figure 6.8: Illustration of the average universal transformation matrix $\overline{M_p^{50}}$

times with $\overline{M_p^{50}}$ to obtain the new state of nodes in each shell at the 351th time point and once with $\overline{M_p^{12}}$ to get the new state of nodes at the 363th time point and once with $\overline{M_p^{3}}$ to reach the time point we requested. If we simulate it with the methods in refinements 1 and 2, we have to repeat the algorithm 6.4 365 times with $\overline{M_p^{1}}$. Compared with the new method, these methods cause more deviations, since there are more repetitions (365 times repetitions for old methods and 9 times repetitions for new method). The theoretical runtime is as follows: for the chosen matrices that we need for the algorithm, we only need to calculate them once and the according theoretical runtime is $((t - t_1) + (t - t_2) + \ldots + (t - t_m)) \cdot (n + 2(s + 1)^2) \subset O((mt - t_1 - t_2 - \ldots - t_m) \cdot (n + s^2))$, where t is the total amount of the observation time points, t_i (*i is the chosen time point*) is the chosen time point for the average universal transformation matrix with steps, m is the number of chosen time points and s is the maximal shell. The runtime for the algorithm is $O(R(s + 1)^2) + O(R(s + 1)) \subset O(Rs^2 + s)$, where R is the times of repetitions and s is the maximal shell.

6.3 III. Attempt

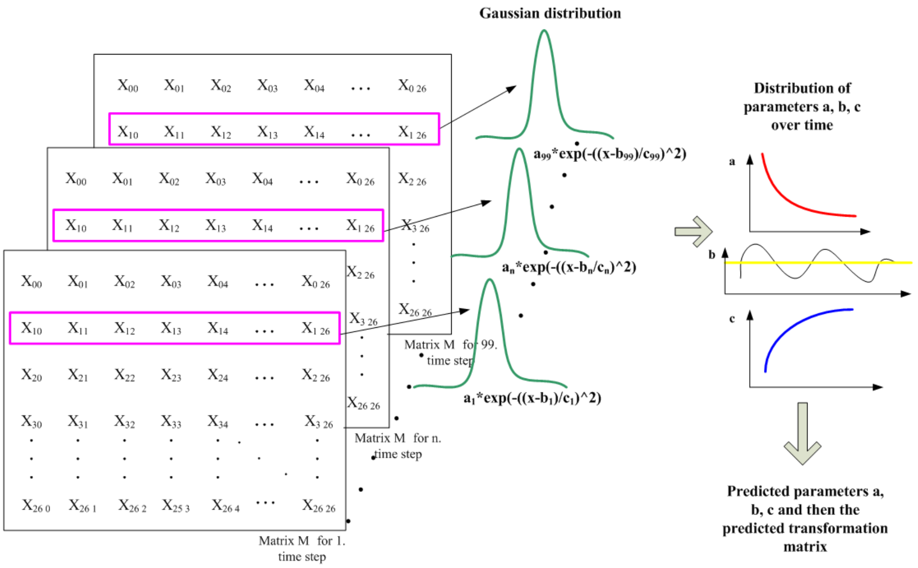

Gaussian distribution
X00 X01 X02 X03 X04 ... X0 26
X10 X11 X12 X13 X14 ... X1 26
X20 X21 X22 X23 X24 ... X2 26
X30 X31 X32 X33 X34 ... X3 26
X26 0 X26 1 X26 2 X25 3 X26 4 ... X26 26
a99*exp(-((x-b99)/c99)^2)
an*exp(-((x-bn/cn)^2)
a1*exp(-((x-b1)/c1)^2)
Matrix M for 99. time step
Matrix M for n. time step
Matrix M for 1. time step
Distribution of parameters a, b, c over time
a
b
c
Predicted parameters a, b, c and then the predicted transformation matrix

Figure 6.9: Illustration of matrices

All nodes states (or shellness) are averaged with window size 29. In the modeling process of our second attempt, we also found that each row of the transformation matrix for a time step M_p^1 can be fitted into a Gaussian distribution (see the illustration in Fig 6.9). One thing we would like to

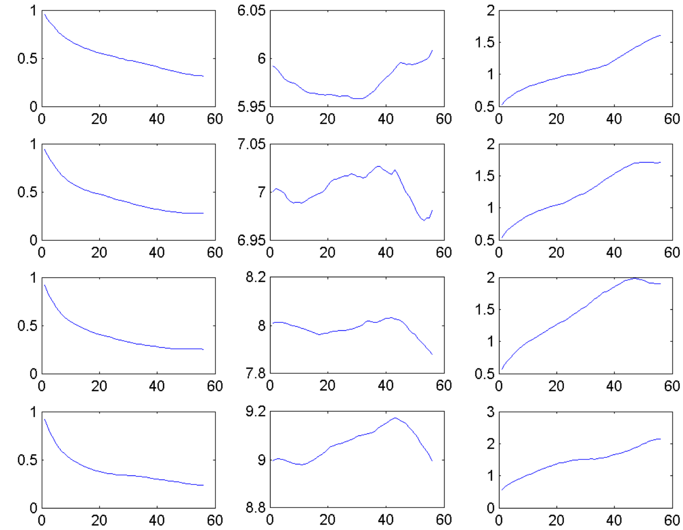


Figure 6.10: The example of distributions of the parameters a, b, c for different rows (shells)

mention, that addition to the R environment, we also use the tool Matlab in this and the next attempt to accomplish the curve fitting process.

It means we can calculate the value of each entry in the row according to the curve in Gaussian form, $a \times e^{-(\frac{x-b}{c})^2}$. The curve is decided by three parameters a, b and c. Furthermore, the Gaussian curve for a row evolves over time. As we can see, if we can predict the value of parameters a, b and c at the according time point, we can then calculate the row of the transformation matrix for the according time point and likewise for all other rows in the transformation matrix too. So that we can obtain the transformation matrix M_G for the according time point and simulate the Internet topology at the requested time point.

We calculated the Gaussian curves for a row over the whole time: $a_i \times e^{-(\frac{x-b_i}{c_i})^2}$, $i = 0 \cdots 26$ and we found approximately that over the time the distribution of the parameter a is a two term exponential distribution. The value of parameter b fluctuates around an average value. And the distribution of the parameter c is a power series model distribution. Fig 6.10 is an example of distributions of the parameters a, b, c for different rows (shells). Every row of this figure is the distribution of the parameters a, b, c for a shell.

Now we can tell the values of parameters a, b and c for each row of the transformation matrix M_G at any time point and then calculate the new state of nodes in each shell by $NewState = f(StartState, M_G)$ (the function f is the same function as in 6.4).

In this attempt, we tried to use a more mathematical way to simulate the Internet topology. However, we found in our experiments, the nodes percentage of each shell moving to all shells is not very close to the real Internet topology, except for the percentage of the nodes staying at their own shell for a time step, namely the value of entries $x_{i,i}$ $(i = 1 \cdots 26)$ in the transformation matrix. Because the value of other entries or the percentage of changes, such as $x_{2,1}$ (percentage of nodes in the shell 2 moving to the shell 1), is very small compared with the value of $x_{i,i}$ and any tiny difference between the simulated curve and the real value of the real Internet would cause the imprecision. You can also see this in 6.5.2.

6.4 IV. Attempt

In our research of Gaussian curve model we realized two things. Firstly, the curving fitting for the values of $x_{i,i}$ in the transformation matrix is precise and successful. Secondly, too many mathematical constraints, such as Gaussian curve, on the other entire, whose value is small, in the transformation matrix is unsuccessful, because any tiny difference causes big imprecision. Hence, we thought of an alternative way to simulate the Internet topology using curve fitting with the refinement we used in attempt 2.

Fig. 6.11 is an illustration some idea of our model. We firstly calculated average universal transformation matrices with step i, $\overline{M_p^i}$ $(i = 1 \cdots 55)$, described in 6.2.5. Using curve fitting we found the distribution of the value of each $x_{j,j}$ $(j = 1 \cdots 24)$ in $\overline{M_p^i}$ $(i = 1 \cdots 55)$ is a two term exponential distribution, in the form of $y = ae^{bx} + ce^{dx}$ $(a, c > 0\ and\ b, d < 0)$. For example, the distribution of the value of the entry $x_{2,2}$ in matrices $\overline{M_p^1}$, $\overline{M_p^2}, \cdots, \overline{M_p^{55}}$ is a two term exponential distribution. Overall we have 24 curves for entries $x_{i,i}$ $(i = 1 \cdots 24$, the results for the curving fitting for entries $x_{25,25}, x_{26,26}$, also known as the maximum shells 25 and 26, is not good, so we abandoned them). And along these curves we can tell and predict the diagonal of any

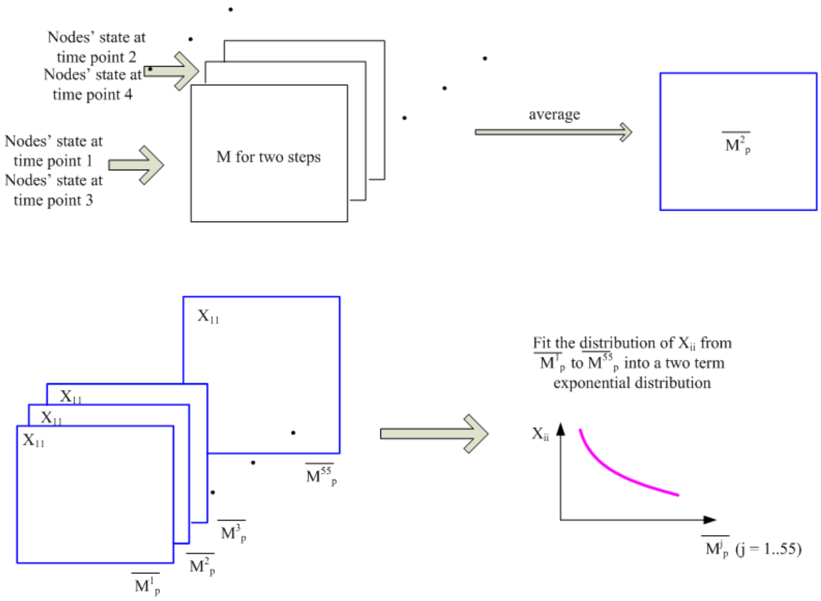

Nodes' state at time point 2
Nodes' state at time point 4
Nodes' state at time point 1
Nodes' state at time point 3
M for two steps
average
$\overline{M^{2}_{p}}$
X_{11}
X_{11}
X_{11}
X_{11}
$\overline{M^{55}_{p}}$
$\overline{M^{3}_{p}}$
$\overline{M^{2}_{p}}$
$\overline{M^{1}_{p}}$
Fit the distribution of X_{ii} from $\overline{M^{1}_{p}}$ to $\overline{M^{55}_{p}}$ into a two term exponential distribution
X_{ii}
$\overline{M^{j}_{p}}$ (j = 1..55)

Figure 6.11: Illustration of the matrices

average universal transformation matrix.

Algorithm 7: To simulate the Internet topology at the AS level at the requested time point

```
Data: Start state of an Internet topology, the time point of the
      simulated Internet topology
Result: Simulated Internet topology at the according time point
$NodesTempState1 \leftarrow NodesStartState$ ; /* The state here is every node's shellness */
for $i \leftarrow 2$ to the request time point do
    $NodesTempState2 \leftarrow g(NodesTempState1,\ \overline{M^1_p})$ ; /* function g changes the current nodes' state to the next (new) nodes' state with the transformation matrix */
    Calculate the average universal transformation matrix $\overline{M^{i-1}_p}$ with $NodesStartState$ and $NodesTempState2$;
    Set the diagonal of $\overline{M^{i-1}_p}$ according to the fitted curves and adjust the rest entries in the same row of $\overline{M^{i-1}_p}$ so that the sum of same row equals 1;
    $NodesTempState1 \leftarrow g(NodesStartState, \overline{M^{i-1}_p})$;
return $NodesTempState1$
```

Algorithm 7 is the main idea of our method. Now we use the function g and the average universal transformation matrix with step 1, $\overline{M_p^1}$, to calculate the new state of nodes (every node's shellness) at time point k. Comparing the new state with the state at the first time point, we can obtain the average universal transformation matrix with step $k-1$, $\overline{M_p^{k-1}}$. We set the diagonal of $\overline{M_p^{k-1}}$ as the same of the diagonal calculated according to the exponential curve and adjust other entries in the same row as the diagonal entries in the matrix, in which, according to the original proportion of each entry in the same row, we distribute the difference between the set value of $x_{i,i}$ $(i = 1 \cdots 24)$ and the original value in the matrix so that the sum of each row equals 1. For instance, if a row of the matrix is $0.1,\ 0.2,\ 0.3,\ 0.4$ and now we set the want to set the second entry to 0.1. Then overall we have 0.1 more. We distribute this 0.1 to all entries according to their original proportions and the row should be $0.11,\ 0.12,\ 0.33,\ 0.44$. We refer to this process as the *adjustment*. We repeat the above calculation and adjustment until we obtain the average universal transformation matrix which we want, such as $\overline{M_p^{100}}$ for the new state of nodes at time point 101. Then we can tell the new state of nodes at time point 101. The theoretical runtime for this algorithm is $O(nt) + O(t(s+1)^2) + O(nt) + O(nt) + O(t(s+1)^2) + O(t(s+1)^2) + O(t(s+1)^2) + O(nt) = O(4nt) + O(4t(s+1)^2) \subset O(nt + s^2t)$, where n is the amount of nodes, t is the time point and s is the maximal shell. The two $O(nt)$ parts are for the assignments in line 1 and line 6 in the Algorithm 7. The two $O(t(s+1)^2) + O(nt)$ parts are for the g function used in line 3 and 6. And the two $O(t(s+1)^2)$ parts are for the calculation of the matrices and the process of the adjustment in the algorithm.

6.5 Evaluation of Different Simulation Methods

In this section we would compare all the methods that we mentioned above simulating the Internet topology at the AS level concerning two aspects.

1. The precision of the amount of nodes in each shell simulated by our model compared with the amount of nodes in each shell of the real Internet topology
2. The dynamics of nodes in each shell, namely, the percentage of node moving from one shell to another shell of simulated topology with the real Internet.

6.5.1 Normalization of the observation data

If we only choose the real Internet topology at a time point, it is too special. We cannot ensure that it is the common Internet topology. In addition, we also would like to focus on the common trend of a node's shellness preference of the real Internet topology. Therefore, before the comparison, we "normalized" our observation data of the Internet. We have the data information of the Internet topology at 100 time points, and also for 99 time steps. Firstly, we averaged each node's state (shellness) with window size 29. It assures us the common trend of each node's shellness preference. For the 99 time steps, we calculate the matrix $\overline{M_p^i}$ $(i = 1 \cdots 55)$ described in 6.4, so that we obtain the average percentages of nodes in one shell moving to all shells for time points $1 \cdots 55$ (matrix $\overline{M_p^1}$ for time point 2, $\overline{M_p^2}$ for time step 3 and so on).

6.5.2 Comparison of Different Simulation Methods

Comparison of the Amount of Nodes in Each Shell

We chose three methods which we think the best threes simulating the Internet, the method in attempt 2 refinement 2 (6.2.4) (average window size 29 and average universal transformation matrix $\overline{M_p}$), the method in attempt 2 refinement 3 (6.2.5) (average universal transformation matrices with different steps $\overline{M_p^i}$ $(i = 1,\ 3,\ 6,\ 12,\ 25,\ 50)$)and the method in attempt 4 to simulate the Internet topology at the AS level at time point 70 and 215. We calculated the amount of nodes in each shell for every simulated topologies. And we compared the amounts of nodes in each shell at time point 70 with the ones of the real Internet and the normalized Internet at the time point 70. The time point 215 is in the middle of year 2008, we can prove the precision of our models in some future work.

Figure 6.12 and 6.13 are the illustrations of the comparison between the real Internet, the normalized Internet and the simulated Internet topology at the time point 70 and 215. The x coordinate denotes shell 1 to shell 26 and the y coordinate illustrate the amount of nodes in the according shell. You can also see the data table in Appendix A-2.1. As we can see in 6.12, the amount of nodes in each shell is very close to the one of the real Internet or the normalized Internet. According to our statistics, the deviations of the number of nodes in each shell of simulated topology compared with the real one are only about 5% for each shell. The nodes' distribution in each shell of the simulated topologies are also very similar to the real ones. However, we should also notice that the nodes' distribution in shell 25 and 26 is reverse. It's a difference between the simulated topologies. Actually, the amount of nodes in the maximal shell should be more than the one in the $max-1$ shell. However, in our observation, shell 26 does not exist all the time. Sometimes, the maximal shell is shell 25. Therefore, in our average value, the amount of nodes in shell 26 is less than the one in shell 25.

Comparison of the Percentage of Nodes of a Certain Shell Moving to Another Shell

We illustrated the comparison of amount of nodes in each shell in the last part. In this part we will compare all our models (or methods) concerning the nodes' dynamics in each shell over time. Figures 6.14, 6.15, 6.16 are some illustrations of comparisons of how nodes move among shells of the topologies of the real Internet, the normalized Internet and the ones simulated with our different methods. You can also find some example comparison figures in Appendix A-2.2. The x coordinate denotes time points (from 15 time point to 70 time point of our observations, because there are some jitters in the first 15 time points). The y coordinate indicates the percentage of nodes in a shell. It shows us, at a certain time point how many percentages of nodes in a shell moving to another shell or staying at their own shell. The different colors of lines in the figures denotes simulated topologies using different methods, the real Internet and the normalized Internet. It is described as follows:

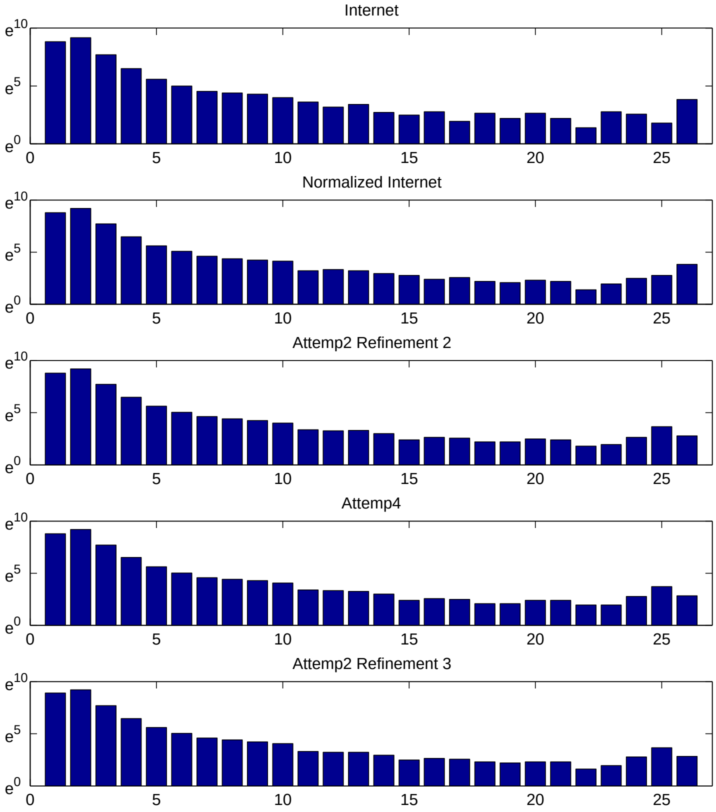


Figure 6.12: Comparison of amount of nodes in each shell at time point 70

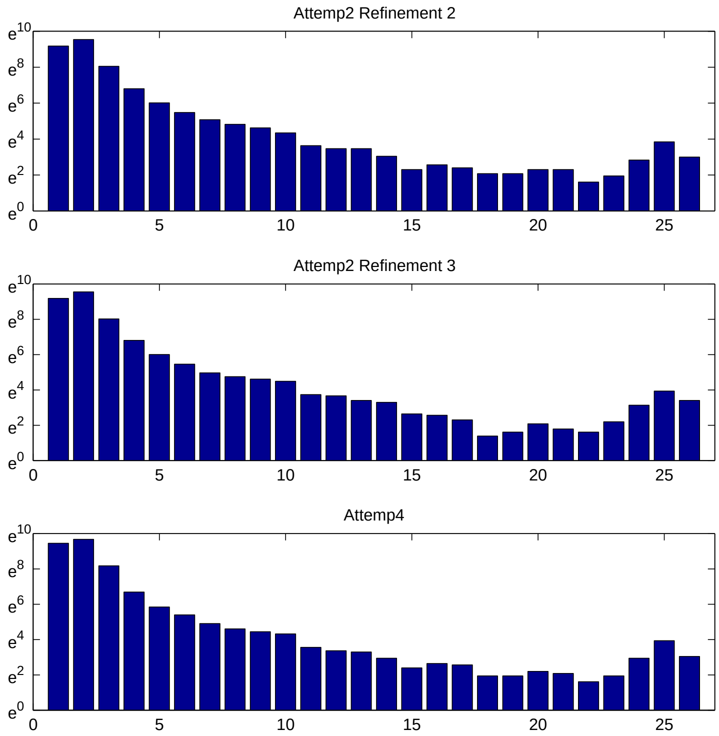


Figure 6.13: Comparison of amount of nodes in each shell at time point 215

1. Black line: the real Internet with average window size 29
2. Red line: the normalized (with average window size 29) Internet
3. Yellow line: the normalized (with average window size 9) Internet
4. Dark blue line: the simulated topology using method in attempt 2 with average window size 9
5. Green line: the simulated topology using method in attempt 2 with average window size 29
6. Blue circles: the simulated topology using method in attempt 2 with different average transformation matrices with steps
7. Pink line: the simulated topology using method in attempt 3
8. Lake blue line: the simulated topology using method in attempt 4

Fig. 6.14 shows us in each shell what percentages of nodes, compared with the state at the start time point, stay at their own shell over time.

For example, in the real Internet and the normalized Internet at time point 20, 80% nodes, which were in the shell 1 at the first time point, are now staying at the shell 1 and 87% nodes , which were in the shell 2 and now are staying at the shell 2. Fig. 6.15 illustrate the proportion of new nodes, compared with the state at the start time point, added into each shell over time. For instance, at time point 20, 10% nodes in the shell 1 are the new added nodes (at the first time point) and 8% nodes of the shell 2 are the new added nodes. Fig. 6.16 tells us the proportion nodes, which move one shell down, in the present shell (also compared with the state at the start time point). For example, 5% nodes of the shell 2 are the nodes moving from the shell 1 at the start time point.

Apparently, the line which is more closer to the red (or the black) one is the most precise method simulating the Internet topology in this part. As we can see, no matter in which case, the green line is more closer to our reference Internet data than the dark blue line. The values of the blue circles are more precise than the green line. It proves again the effects of refinements in the algorithm 2. The Gaussian model is OK in low shell area. Sometimes it is more precise than other methods (see 6.14, the pink line is almost identical with the red line in low shell area). But in the high shell area above the shell 20 it is a little messy and imprecise. We can also see, the lake blue line is very accurate in Fig 6.14, because in our model, we set the diagonal of the matrix according the data of the normalized Internet. Furthermore, the lake blue lines in other comparison figures are also very close to the reference Internet data.

In our opinion, the model with the adjustment in attempt 4 is so far the best in our proposed models. First of all, it very precisely distribute the nodes, which stay at their own shell over time. As we already know, the nodes stay at their own shell is a part of the whole nodes' movement (see 5.2.6. This model for all other situations, such as nodes moving one shell higher or downer etc., is also accurate. Sometimes it might not be as accurate as the model in attempt 2 using 6 average universal matrices with different steps or the one using average universal matrices with average window size 29, but they are only small differences and the big part (nodes staying at their own shell) is very precise. In addition, it combines mathematical methods (curve fitting) with the real data information together. Therefore, we think this model is our best model. The second best model, from our point of view, should be the method in attempt 2 using 6 average universal matrices with different steps. From all the figures, we can find it is more precise than other left methods and it is simple. The third successful model is the method in attempt 2 using average universal matrix with average window size 29. The Gaussian model we proposed is not very successful, but it tried to simulate the Internet more mathematically. Hence, it is a good try-out.

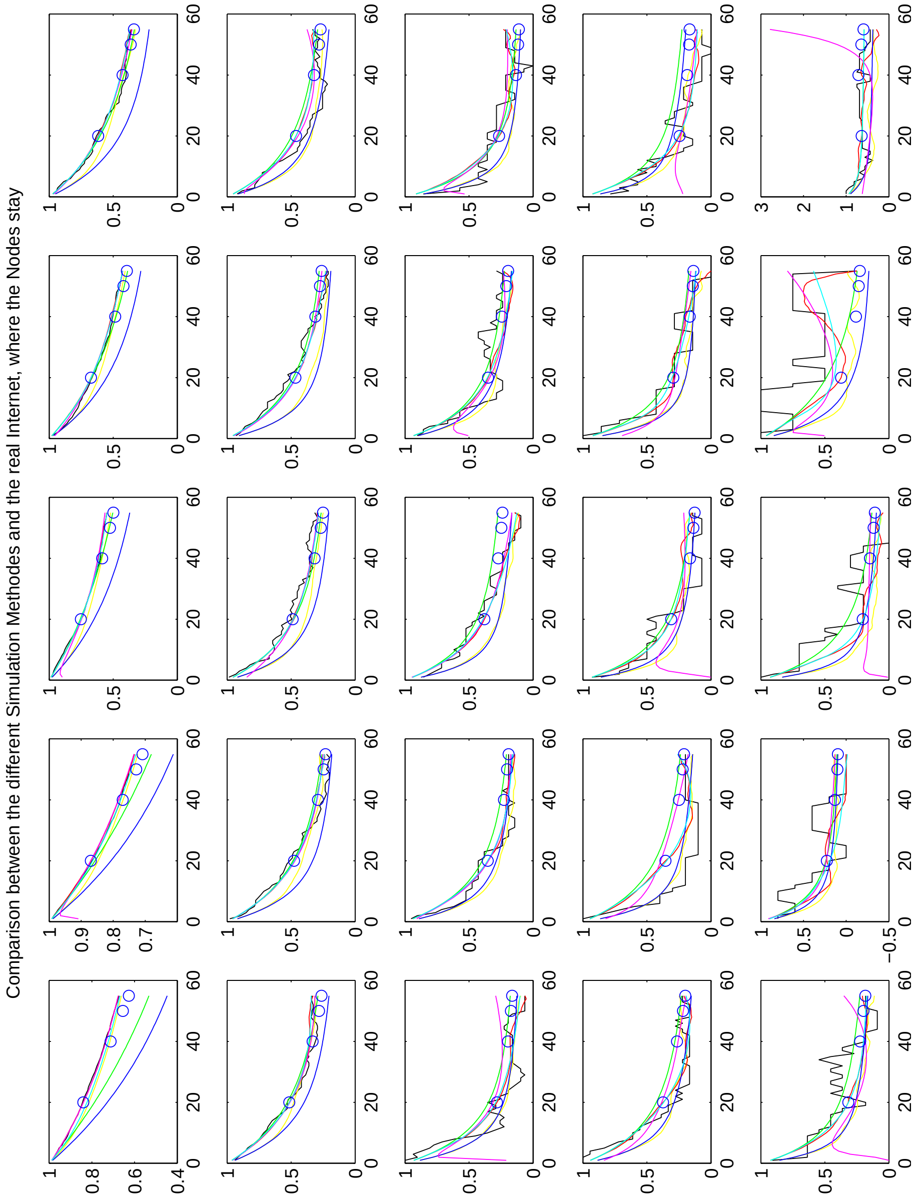


Figure 6.14: Comparison of the percentages of nodes stay at own shell over time between the simulated topology and the normalized Internet topology

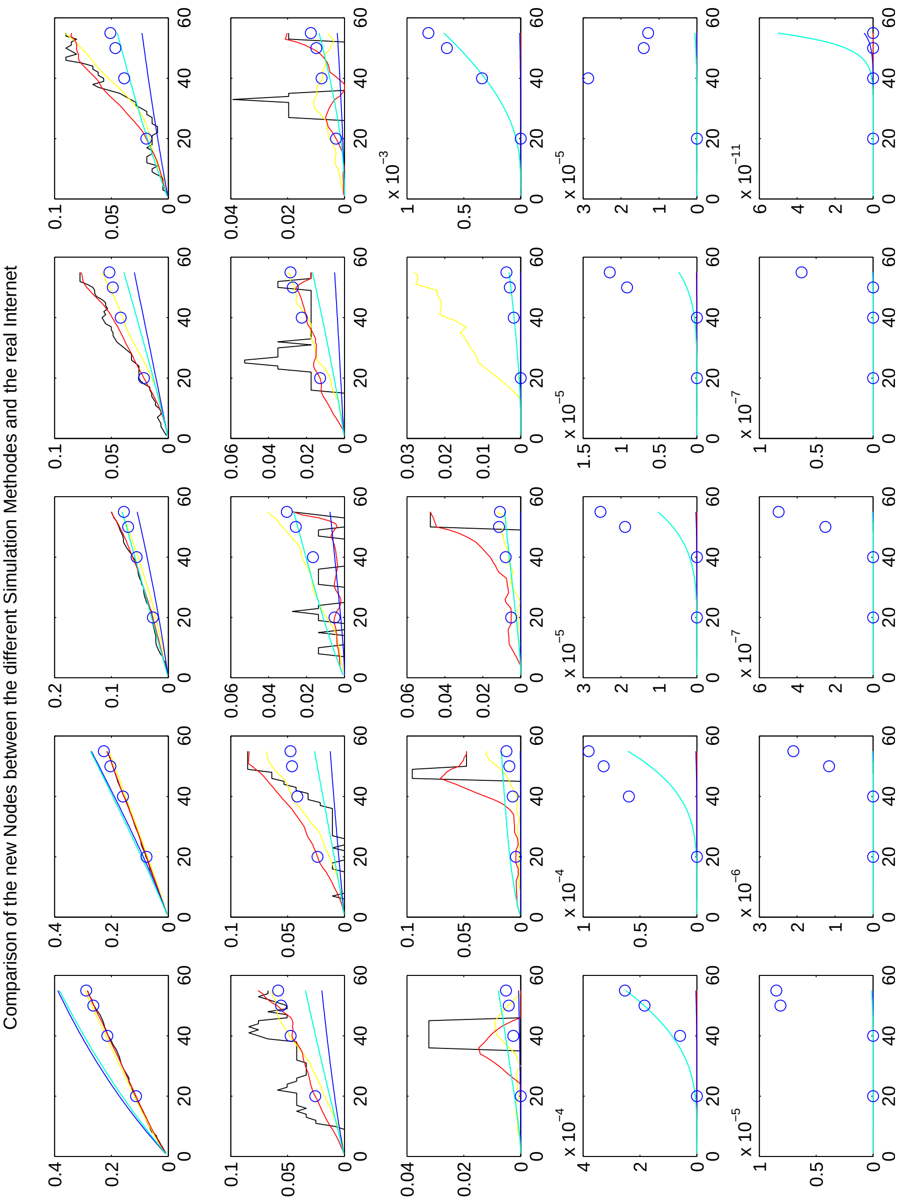


Figure 6.15: Comparison of the percentages of new nodes added into each shell over time between the simulated topology and the normalized Internet topology

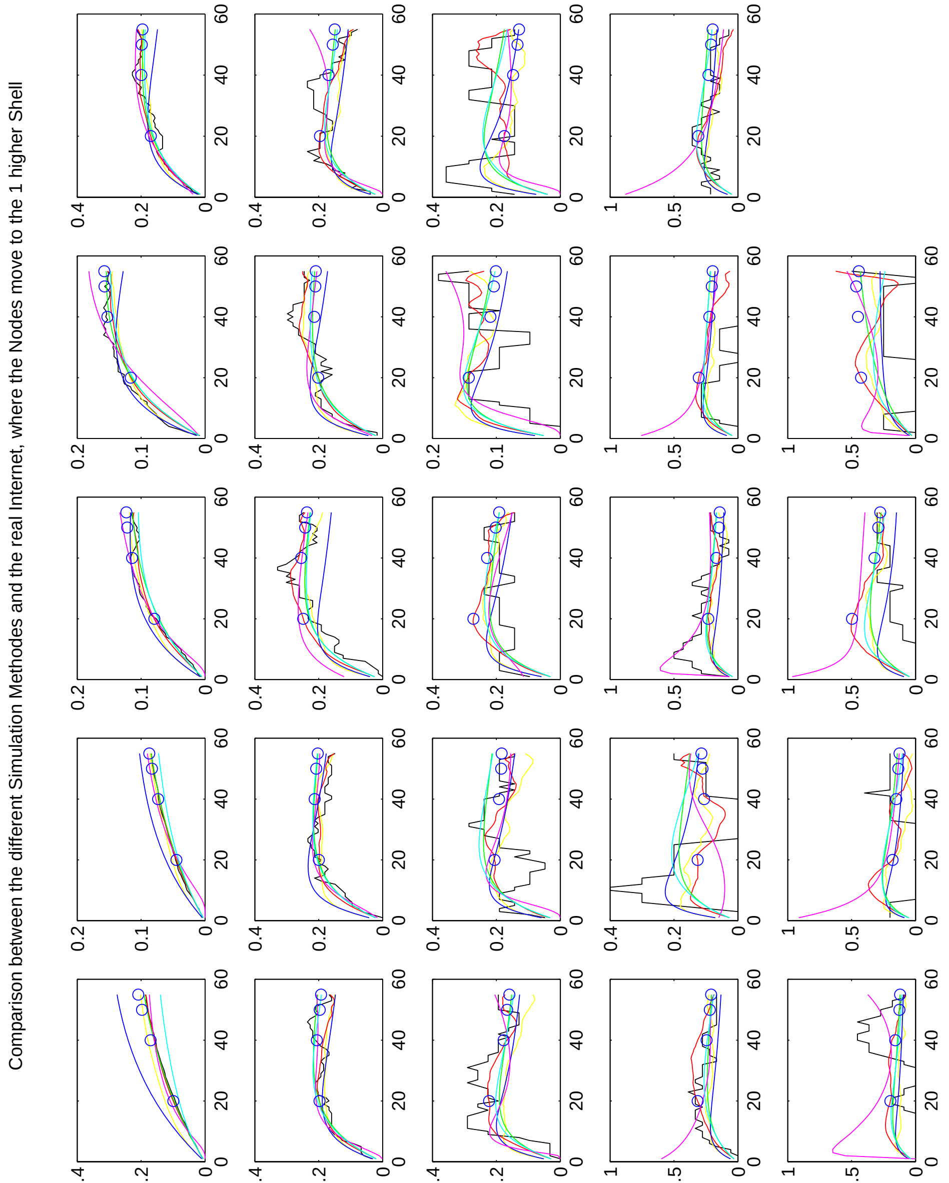


Figure 6.16: Comparison of the percentages of nodes moving to one shell higher over time between the simulated topology and the normalized Internet topology

6.6 Summary

In this chapter, basing on the data we researched in the last chapter, we tried different ways to model the evolving Internet topology at the AS level. We propose and evaluate four models to simulate the Internet topology and three of them arerelatively successful : the adjustment model (6.4), the model using different average universal matrices with different steps (6.2.5), and the model using average universal matrix $\overline{M_p}$ with average window size 29. In our opinion, the adjustment model is so far the best method in our proposed models, since in the comparison (6.5) it is precise and it combines mathematical methods (curve fitting) with the real data information together. We also proposed a Gaussian model that used a mathematic distribution to model the Internet topology. However, it is only precise in the low shell area. but it tried to simulate the Internet more mathematically. Hence, it is a good try-out.

Chapter 7

Conclusion

In this work, we observed two different network topologies according to some popular metrics and analyzed them in an evolutionary way. We revealed some regulations how these two networks evolve over time. We researched the *dm* chemist supermarket and compared our visualized results with our statistic results to ensure the feasibility of our approaches to analyze an evolving network concerning a node's dynamics in the network over time. And then we applied our approaches to research the Internet topology at the AS level. Furthermore, on the basis of our analysis, we proposed different ways to model the Internet at the AS level by simulating its evolution over time. In this chapter, we summarize our achievements and outline the work that can be focused on in the future.

7.1 Achievements

We chose some different popular metrics (degree, frequency, core, rich club connectivity etc.) and calculated these values of the *dm* receipt-product-receipt network (every month from Oct. 2004 to Oct. 2006) and the Internet topology at the AS level (every seven days from Apr. 2004 to Feb. 2006). We analyzed these two network topologies with calculated values and observed them, how they evolved over the whole time.

1. For the *dm* receipt-product-receipt network, we found that the degree distribution of nodes of the p-r-p network has a steady structure over time. The trade of products is also stable. Some certain products are always bought together. There are 13% products, such as plastic bags and kitchen paper, that always stay in the maximal shell over time, which means they are always sold a lot over the whole time. A popular product of a month wouldn't get sold more the next month.

The customers have certain shopping habits. They always buy a lot of things, which are used commonly in our daily life, and the products that are durable or rarely used are always sold less. And they also have some certain consuming seasons, such as December and July are two best seller seasons meanwhile in February products are sold least. The core products, e.g. daily used stuffs, are always bought together, since they have a stable high rich club connectivity value. It implies again that the core of the p-r-p network is steady and the customers have certain shopping habits. The nodes' distributions over time (including the new added nodes) are pretty chaotic: a node in a high shell could suddenly fall into a low shell, because it's sold less or a new added node could be in a very high shell at the first time, since it's popular. It depends on the own feature of the product. We compared our visualized results with our statistic results to ensure the feasibility of our approaches to analyze an evolving network concerning a node's dynamics in the network over time.

2. For the Internet topology at the AS level, we found that the Internet topology has a steady structure. All the degree distributions are similar over the whole time. Nodes with low degree (degree 1 and 2) constitute the majority of the network. Over half of the nodes have no degree changes at all. Most of the rest have small degree changes and these changes took place in the low degree groups. Only a few ASes in the backbone group have some dramatical degree changes. Compared with the first time point, there are altogether 6472 new nodes added into the network and 97% of them are with low degree (< 5). Therefore, the degree of most nodes increase slightly. Besides, the number of the nodes with high degree (backbone ASes) stay quasi the same over the whole time.

   The nodes in the shell 2 and the shell 1 are the majority of the whole Internet (over 70% of the nodes) and there are more nodes in the shell 2 than in the shell 1. Most (over half) of the nodes have small shellness changes. The nodes which have bigger shellness changes are the ASes, such as operators, in the backbone of the network. Over half of the nodes stayed in the original shell and the rest of them moved to a shell near their original shell (they go 1-2 shells up or down). Only a few nodes moved dramatically among shells. The majority of the maximal shell is composed of the nodes (87% of them) which are originally in the maximal shell at the first time point. The small rest part is composed of the nodes moving from shell 25-16 over time. If a new node (compared with the first time point) is added into the network, it participate firstly in the low shell. As time goes by, it will stay in its shell or move to a higher shell according to its feature of its own, for instance, if it is a backbone AS, it starts with low shell and will eventually move into a very high shell. The quantity of the nodes in the Internet at the AS level is a linear growth (except the jitter in May. 2004). The growth of the amount of the nodes in each shell is also quasi linear.

We also found some relationship between the degree of a node and the shellness of it. A part of the nodes with degree k are distributed in the shell k, meanwhile the other part of the nodes are distributed in the shell $k-1$, $k-2$, $\cdots$, 1. As time goes by, despite a lot of new nodes with different degrees were added into the Internet, the distribution proportion for different degrees in a shell always stay quasi the same.

Besides above mentioned points, we also found that the value of the rich club connectivity decreased by 1 every year from the year 2004 to the year 2006. We assume that this value decreases by 1 every year until it get some kind of balance. However we need more observation data for this assumption in the future.

3. On the base of our observation, we tried different ways to model the evolving Internet topology at the AS level. We proposed three relatively successful evolving models for the Internet topology: the adjustment model (6.4), the model using 6 average universal matrices with different steps (6.2.5), $\overline{M_p^1}, \overline{M_p^3}, \overline{M_p^6}, \overline{M_p^{12}}, \overline{M_p^{25}}, \overline{M_p^{50}}$ and the model using average universal matrix $\overline{M_p}$ with average window size 29. In our opinion, the adjustment model is so far the best method in our proposed models, since in the comparison (6.5) it is precise and it combines mathematical methods (curve fitting) with the real data information together. We also proposed a Gaussian model that used a mathematic distribution to model the Internet topology. However, it is only precise in the low shell area, but it tries to simulate the Internet more mathematically. Hence, it is a good try-out.

7.2 Scope

7.2.1 Several points could be done in the future

This work has achieved some new analysis results in the research of modeling the complex network. However, because of the limitd time, there is still much work that could be done in the future.

1. In our observation we found that the value of rich club connectivity for the Internet topology at the AS level decreased by 1% each year, but we also need more observation data in the future for the assumption.

2. How the network topology evolves, such as the *dm* product-receipt-product network and the Internet topology at the AS level, actually depends on the own features of a node. For instance, in our work, how is a product distributed in the p-r-p network is decided by its popularity. If it is popular, then it is bought a lot with other products and consequentially it has a high degree and is distributed in a high shell. Or whether an AS has a high degree and is in a high shell, is influenced by that if it is a backbone AS. In (DM00) and (Bar00), such a situation is also proposed. The different features of a node influence the evolution of a complex network. In the future we can also research different features of a node to model the network better.

3. We can combine the proposed evolving model for the Internet at the AS level with the Core Generator from our institute to simulate the Internet topology.

7.2.2 New Aspects for the Modeling of the Complex Network

Although the history of the research on complex networks is not long, it is becoming an important new subject. In the past few years, a lot of achievements have been reached in the modeling of complex networks. However, there are still some other areas for the researchers to work on.

1. **Directed network.** The WWW network, the citation network and many other important real networks are actually directed networks. Except for some basic features, we know few about the features of the directed network. In the directed network, it can not be assured that all other nodes are reachable from one identified node, which leads to that in the network there are some small groups depending on the start nodes. Moreover, the indegree and the outdegree of a node in the directed network is different. Most current models of the complex network ignore this feature of the network. Hence, it is a challenge to find an evolving model of directed network.

2. **Accelerating network.** In the current evolving models for complex networks, the growth of the quantity of the nodes and edges is linear. However some researches show us that the increment of the number of edges in the WWW network, the Internet etc. is faster than the one of nodes. Such networks are called the accelerating network (MG). To analyze the influence of the accelerating increment on the feature of the network topology is another aspect.

3. **Self-similarity network.** In the real system, lots of network topologies have the feature of self-similarity (SHM05). What is the mecha-

nism to generate the self-similarity structure? What are factors that decide the self-similarity feature? This is also a new aspect to challenge.

Bibliography

[AANa] Reka Albert, Istvan Albert, and Gary L. Nakarado. Structural vulnerability of the north american power grid. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0401084, year = 2004.

[AANb] Reka Albert, Istvan Albert, and Gary L. Nakarado. Structural vulnerability of the north american power grid. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0401084, year = 2004.

[AB00] Reka Albert and Albert-Laszlo Barabasi. Topology of evolving networks: local events and universality. *Physical Review Letters*, 85:5234, 2000. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0005085.

[AJB99] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. The diameter of the world wide web. *Nature*, 401:130, 1999. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9907038.

[AJHAdS05] J. S. Andrade Jr., Hans J. Herrmann, R. F. S. Andrade, and L. R. da Silva. Apollonian networks: Simultaneously scale-free, small world, euclidean, space-filling and with matching graphs. *Phys. Rev. Lett.*, 94:018702, 2005. e-print: cond-mat/0406295.

[Alb06] David Albrecht. Generieren von graphen mit core-hierarchie. 2006.

[BA99] A-L Barabási and R Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999. http://www.ncbi.nlm.nih.gov/.

[Bar00] G. Bianconi A. L. Barabasi. Competition and multiscaling in evolving networks, 2000. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0011029.

[BBV04] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. Weighted evolving networks: coupling topology and weights dynamics. *Physical Review Letters*, 92:228701,

2004. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0401057.

[BFNW04] Justin Balthrop, Stephanie Forrest, M. E. J. Newman, and Matthew M. Williamson. Technological networks and the spread of computer viruses. *Science*, 304:527, 2004. http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0407048.

[BM99] Waxman BM. Routing of multipoint connections. In *SIGCOMM*, pages 251–262, 1999. http://citeseer.ist.psu.edu/michalis99powerlaw.html.

[BRV] A-L Barabasi, Erzsebet Ravasz, and Tamas Vicsek. Deterministic scale-free networks, Feb. http://arxiv.org/abs/cond-mat/0107419, year = 2002.

[CAI] CAIDA. Visualizing internet topology at a macroscopic scale.

[CGA04] Francesc Comellas, Fertin G, and Raspaud A. Recursive graphs with small-world scale-free properties. *Phys. Rev. E*, 2004. http://link.aps.org/abstract/PRE/v69/e037104.

[CnP00] Francesc Comellas, Javier Ozó n, and Joseph G. Peters. Deterministic small-world communication networks. *Information Processing Letters*, 76(1–2):83–90, 2000. citeseer.ist.psu.edu/comellas00deterministic.html.

[CnP02] Francesc Comellas, Javier Ozó n, and Joseph G. Peters. Deterministic small-world networks. 2002.

[DGM02] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Pseudofractal scale-free web. *Phys. Rev. E*, 65(6):066122, Jun 2002.

[DM00] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks with aging of sites. *Physical Review E*, 62(2):1842+, 2000. http://dx.doi.org/10.1103/PhysRevE.62.1842.

[ERm60] *The Evolution of Random Graphs*, volume 5. 1960.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999. http://citeseer.ist.psu.edu/michalis99powerlaw.html.

[FHJS02] A. Fronczak, J. A. Holyst, M. Jedynak, and J. Sienkiewicz. Higher order clustering coefficients in barabasi-albert networks. *Physica A*, 316(1):688–694, December 2002. http://dx.doi.org/10.1016/S0378-4371(02)01336-5.

[FYG03] Chung F, Lu L Y, and Dewey T G. Duplication models for biological networks. *Computational Biology*, 10(5):677–688, 2003.

http://www.math.sc.edu/~lu/papers/bio-papers.pdf.

[G.] Corso G. Families and clustering in a natural numbers network. http://link.aps.org/abstract/PRE/v69/e036106.

[HK02] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65:026107, 2002. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0110452.

[JCJ00] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. 2000. http://citeseer.ist.psu.edu/jin00inet.html.

[Jul] Julian Faraway July. Practical regression and anova using r. http://citeseer.ist.psu.edu/642417.html.

[Kas99] Rajesh Kasturirangan. Multiple scales in small-world graphs, 1999. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9904055.

[KE02] Konstantin Klemm and Victor M. Eguiluz. Highly clustered scale-free networks. *Physical Review E*, 65:036123, 2002. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0107606.

[KKR$^+$99] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999. citeseer.ist.psu.edu/kleinberg99web.html.

[KRR$^+$00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 57–65. 41th IEEE Symposium On Foundations of Computer Science (FOCS), 2000. citeseer.ist.psu.edu/501628.html.

[MG] John S. Mattick and Michael J. Gagen. Accelerating networks. http://www.sciencemag.org/cgi/content/full/307/5711/856.

[NW] M. Newman and D. Watts. Renormalization group analysis of the small-world network model. citeseer.ist.psu.edu/318353.html.

[NW99] M. E. J. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332, 1999. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9904419.

[SHM05] Chaoming Song, Shlomo Havlin, and Hernan A. Makse. Self-similarity of complex networks. *Nature*, 433:392, 2005. http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0503078.

[VS] W. N. Venables and D. M. Smith. An introduction to r. http://cran.r-project.org/manuals.html.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998. http://dx.doi.org/10.1038/30918.

[YJBT01] S. H. Yook, H. Jeong, A.-L. Barabási, and Y. Tu. Weighted evolving networks. *Phys. Rev. Lett.*, 86(25):5835–5838, Jun 2001.

Appendix A

A-1 Tables of Time Series Statistics for the Internet

A-1.1 *dm* p-r-p Network

The development of the amount of numbers in the *dm* network	
dynamic average	geometric average development rate
6260.333	0.9997087

The development of the maximal shell	
dynamic average	geometric average development rate
56.95833	0.9955414

The development of the amount of nodes in each degree group			
	degree 1-40	degree 41-200	degree 201-max
dynamic average	5227	937	94
geometric average development rate	1.0006763	0.9948013	0.9920036

The development of the amount of nodes in shells: Shell 1-10, 14, 15, 24, 25, 34, 35, 44, 45, 54, 55, 62, 63				
dynamic average	985	599	429	33
geometric average development rate	1.0028407	0.9997325	0.9981496	0.9995114
dynamic average	269	236	210	184
geometric average development rate	0.9962167	0.9993335	0.9942031	1.0042260
dynamic average	163	152	112	110
geometric average development rate	1.0015446	1.0064436	1.0048717	0.9992215
dynamic average	70	64	56	44
geometric average development rate	0.9982283	1.0030034	1.0343661	0.9991586
dynamic average	26	24	25	17
geometric average development rate	0.9966704	0.9825353	0.9930636	0.9066725
dynamic average	15	7		
geometric average development rate	0.9484106	1.0000000		

A-1.2 Internet Topology at the AS Level

The development of the amount of numbers in the Internet	
dynamic average	geometric average development rate
19510.33	1.002451

The development of the maximal shell	
dynamic average	geometric average development rate
25.42188	0.9995915

The development of the amount of nodes in each shell (Shell 1-26)				
dynamic average	6545	9233	2158	635
geometric average development rate	1.0027759	1.0022657	1.0023326	1.0026799
dynamic average	254	133	95	75
geometric average development rate	1.0037421	1.0028527	1.0037600	1.0006095
dynamic average	66	50	33	23
geometric average development rate	1.0011876	0.9978243	1.0050318	1.0004253
dynamic average	23	19	14	13
geometric average development rate	1.0042325	0.9940745	0.9962269	1.0104556
dynamic average	13	9	9	10
geometric average development rate	0.9951161	1.0019010	0.9970078	1.0033227
dynamic average	10	7	7	12
geometric average development rate	0.9928057	1.0010981	1.0023271	0.9915884
dynamic average	27	25		
geometric average development rate	1.0452490	0.9983640		

The development of the amount of nodes in each degree group				
	degree 1-5	degree 6-30	degree 31-100	degree 100-max
dynamic average	17520	1731	191	67
geometric average development rate	1.002475	1.002239	1.002315	1.001985

A-2 Comparison in the Evaluation

A-2.1 Comparison of Different Simulation Methods Concerning the Amount of Nodes in Each Shell

Amount of nodes in each shell at time point 70					
	Internet	Normalized Internet	Attempt2 refinement2	Attempt4	Attempt2 refinement3
Shell 1	6878	6560	6581	6557	7351
Shell 2	9622	9905	9883	9900	10008
Shell 3	2209	2257	2253	2219	2197
Shell 4	666	648	654	672	635
Shell 5	267	272	279	275	272
Shell 6	147	162	155	151	154
Shell 7	93	101	103	97	98
Shell 8	81	79	82	83	83
Shell 9	73	69	70	73	68
Shell 10	54	63	55	58	58
Shell 11	37	25	29	30	27
Shell 12	24	28	26	28	25
Shell 13	30	25	27	26	25
Shell 14	15	19	20	20	19
Shell 15	12	16	11	11	12
Shell 16	16	11	14	13	14
Shell 17	7	13	13	12	13
Shell 18	14	9	9	8	10
Shell 19	9	8	9	8	9
Shell 20	14	10	12	11	10
Shell 21	9	9	11	11	10
Shell 22	4	4	6	7	5
Shell 23	16	7	7	7	7
Shell 24	13	12	14	16	16
Shell 25	6	16	39	41	39
Shell 26	46	46	16	17	17

Amount of nodes in each shell at time point 215			
	Attempt2 refinement2	Attempt4	Attempt2 refinement3
Shell 1	9726	9684	12753
Shell 2	13984	14028	15892
Shell 3	3137	3045	3532
Shell 4	903	904	801
Shell 5	411	406	345
Shell 6	240	234	221
Shell 7	161	143	135
Shell 8	124	116	100
Shell 9	102	101	85
Shell 10	77	89	75
Shell 11	38	42	35
Shell 12	32	39	29
Shell 13	32	30	27
Shell 14	21	27	19
Shell 15	10	14	11
Shell 16	13	13	14
Shell 17	11	10	13
Shell 18	8	4	7
Shell 19	8	5	7
Shell 20	10	8	9
Shell 21	10	6	8
Shell 22	5	5	5
Shell 23	7	9	7
Shell 24	17	23	19
Shell 25	47	51	51
Shell 26	20	30	21

A-2.2 Comparison of Different Simulation Methods Concerning Nodes' Dynamics

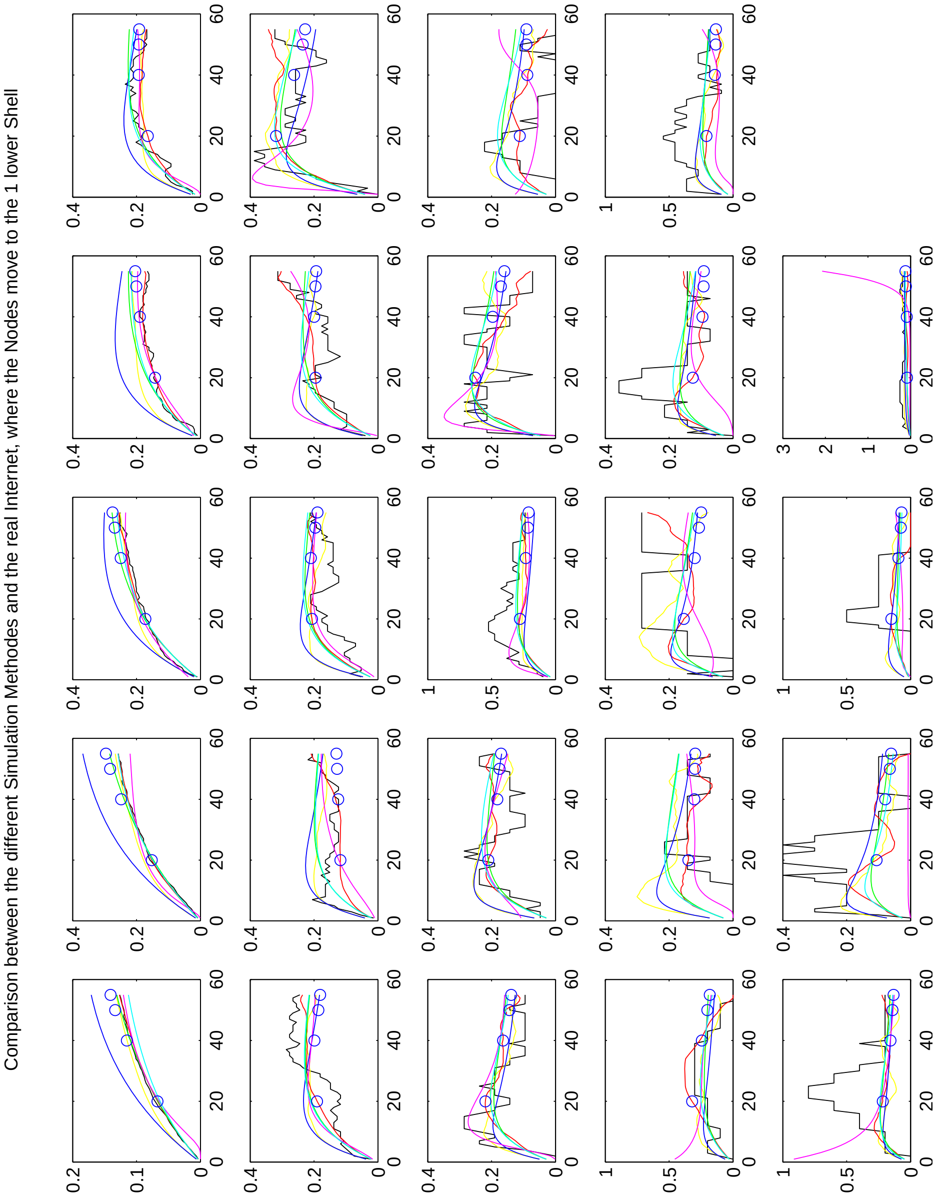


Figure A.1: Comparison of the percentages of nodes moving one shell down over time between the simulated topology and the normalized Internet topology (the different color of lines are the same as described in 6.5.2)

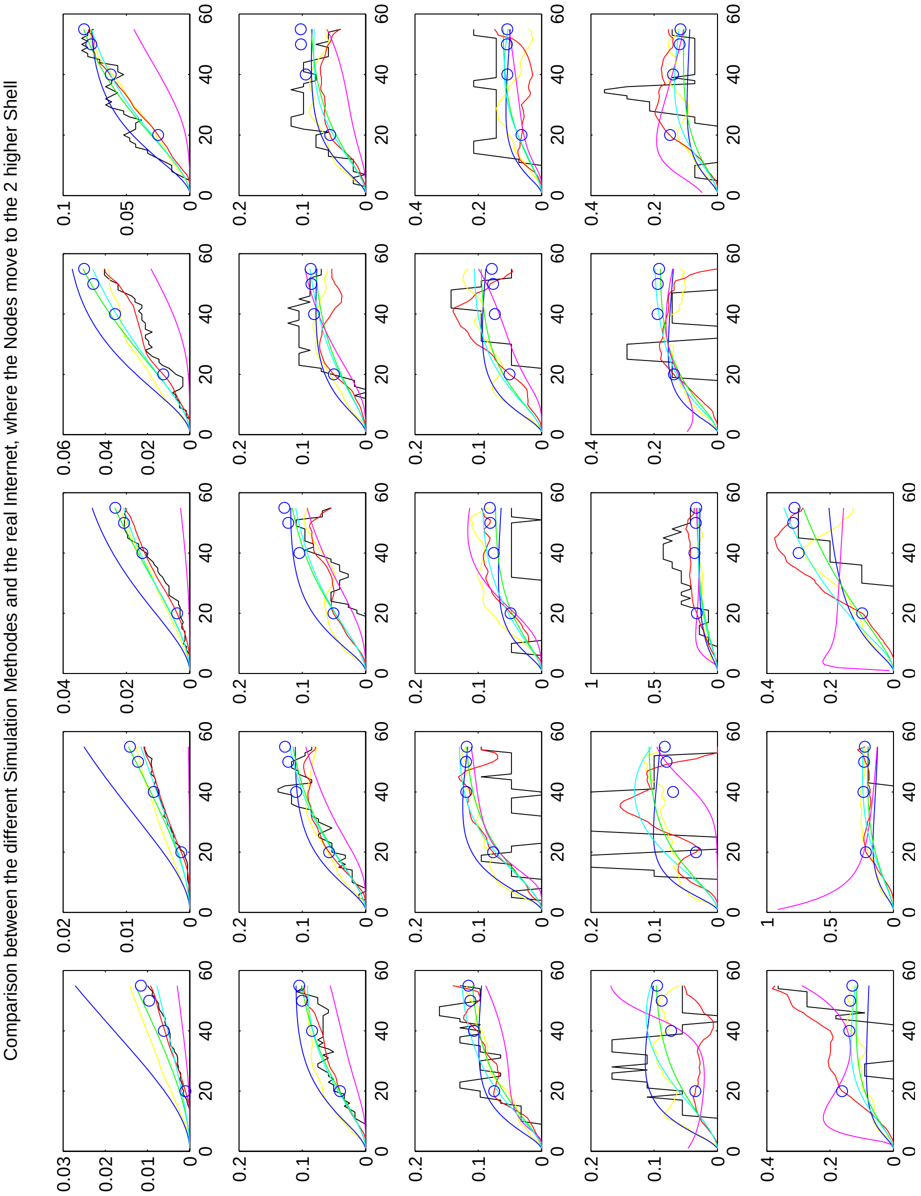


Figure A.2: Comparison of the percentages of nodes moving two shells higher over time between the simulated topology and the normalized Internet topology (the different color of lines are the same as described in 6.5.2)

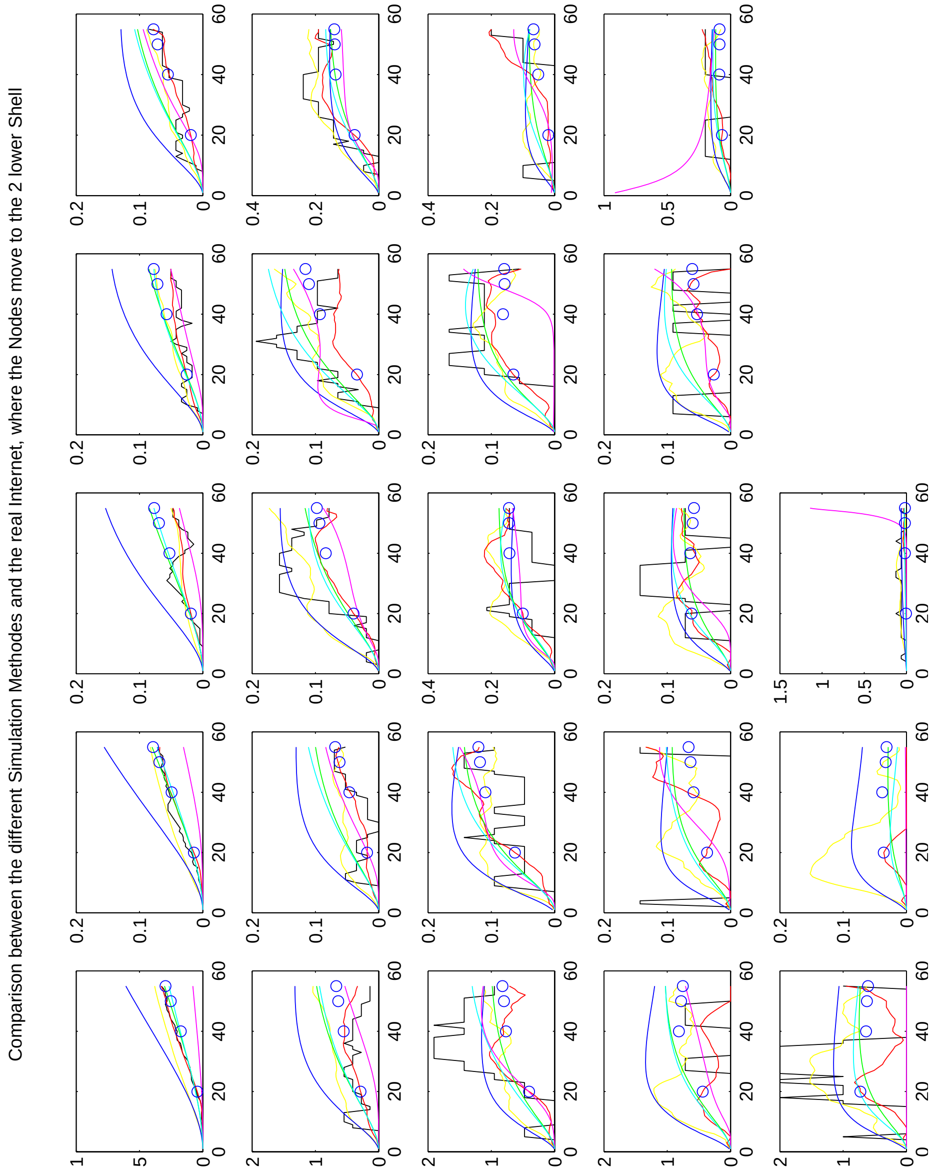


Figure A.3: Comparison of the percentages of nodes moving two shells down over time between the simulated topology and the normalized Internet topology (the different color of lines are the same as described in 6.5.2)

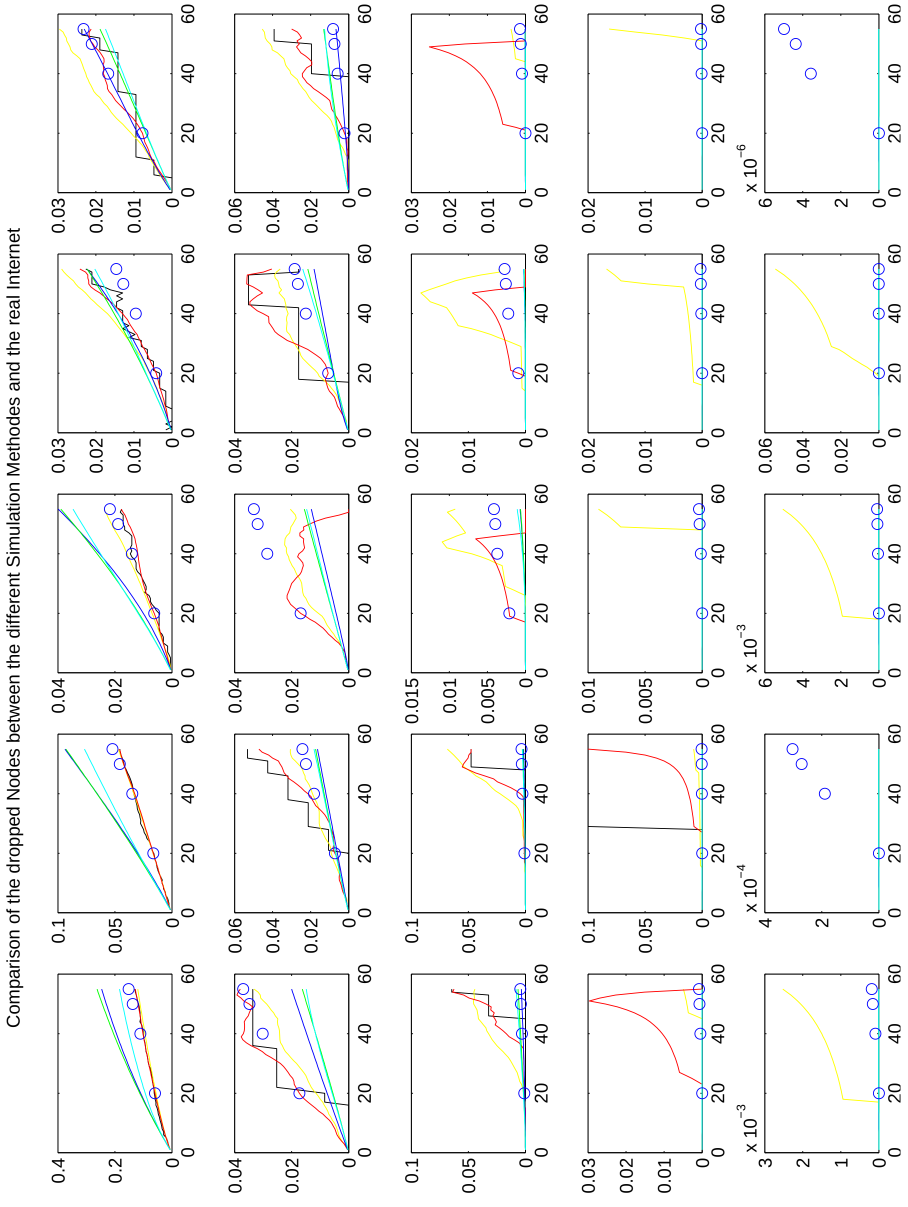


Figure A.4: Comparison of the percentages of nodes dropped from the network over time between the simulated topology and the normalized Internet topology (the different color of lines are the same as described in 6.5.2)