Fachhochschule
Münster University of
Applied Sciences

Single Source Sampling Signature Differences between Random and Scale Free Graph Networks

Master Thesis

Christian Dankbar

Department of Electrical Engineering and Computer Science

Supervising tutors
Prof. Dr. Gernot Bauer
Dr. Markus Loecher

Declaration

Herewith I affirm that I have written this thesis on my own. I did not enlist unlawful assistance of someone else. Cited sources of literature are perceptibly marked and listed at the end of this thesis. The work was not submitted previously in same or similar form to another examination committee and was not yet published.

Rheine, March 2006

Acknowledgement

First of all I would like to thank my supervising tutors Prof. Dr. Gernot Bauer and Dr. Markus Loecher for their abundant help and their prolific suggestions. Especially the support I experienced before and during the educational journey to Princeton, NJ, was outstanding.

I am indebted to the DAAD for the scholarship I obtained for the educational journey to Princeton, NJ.

I dedicate this thesis to my parents who unremittingly supported me during my years of study. They made this work possible.

Table of Contents

Introduction

Complex networks are present everywhere in normal course of life. The Internet is a very good example for a large network in the everyday business of many people. But also social, biological or neuronal networks and many more are examples for complex networks found in everyday life. We can observe a multidisciplinarity of this topic that covers mathematics, sociology, biology, physics and computer science. Originally the exploration of complex networks has been part of graph theory.

A network's topology is the fundament to describe a weblike network. The term complex network describes not a specific network topology but rather the scientific effort to acquire and describe elaborate and sophisticated network structures in a preferable effective way.

A generic network model consists of vertices, also called nodes, and edges, also called links. The links are connections between the nodes. We can virtually "travel" through this network model by starting at any node (our origin), then hopping to a neighbor node, from there to another node and so on, till we have reached our destination node, also called target. The abstract network model can be applied to describe all sorts of networks. To describe the Internet on the router level for example nodes represent computers and routers and edges represent the physical or wireless links connecting them. We also can consider the Internet on the AS (Autonomous System) level. Autonomous Systems are units of router policies under the control of an administrator and all have a unique number. They are based on the IP protocol and all are assigned a specific IP prefix. One AS can be represented by one node, one interlink between two AS then composes an edge in the network model. Considering the World Wide Web we can consider one website as a node and a hyperlink to another website as an edge. We also can describe social networks: one node represents one person and a link is any kind of social interaction to another person. Another example is a biological cell: chemicals are the nodes and chemical reactions are edges linking the nodes. All these examples are only a very little fraction of existent complex networks this model is applicable to.

The simplest topology of complex networks introduced in the 1950s is random graphs. They have been studied by the mathematicians Paul Erdös and Alfréd Rényi. According to their studies, we have N nodes which are being connected in pairs with a probability $0 \leq p \leq 1$ which results in a graph with N nodes and approximately $\frac{pN(N-1)}{2}$ randomly distributed links. Every node has a predictable number of edges. Based on this model there have been many examinations of real networks. These examinations led to the conclusion that real networks cannot only be based on the random network topology. They stated there has to be another, well organized underlying structure. One network topology that has been developed by means of explorations of real networks is the scale free graph. The main characteristic of this topology is that its nodes do not show a typical number of links to other nodes. They are called scale invariant because there is no dominant scale in the degree distribution such as the average degree in Poisson distributions.

Since science has focused on complex networks there has been an effort to acquire or measure complex networks. The best way to explore a complex network is to sample it. Therefore we must have a set of sources from where we start traceroutes to a set of targets. For sampling the Internet on router level this can be done with a program like traceroute. Traceroute tracks the whole route from the source computer it is running on to the given destination by using the ICMP protocol. It returns the IP addresses of every node it has passed on this route, also called hops. All obtained addresses are detected nodes in our network. If we start traceroute several times from the same computer it might occur that the first hops are always the same (same gateway for example). That adds up to a discovery redundancy of these multiply detected nodes. If we start traceroute on more than one computer and target lots of different computers with every traceroute we are able to map the Internet or at least a part of it. This tracerouting procedure is not only applicable for the Internet on the router level but more or less on any real network. For sampling it on the AS level we can use the BGP protocol. Sampling the WWW might be more complicated because there is no specific protocol that provides the data we need. But at least we know every node's true degree because it is obvious how many outgoing links a website has. By contrast we do not know the true degree of a node when sampling the Internet on router level with traceroute because the ICMP protocol does not provide that information how many connections a router has for example. Sampling the social network can be done by a poll. Chemical reactions in cells can be detected by experiments.

By the process of sampling we obtain a subgraph of the scanned original network graph. It is hard to explore the whole network, so the subgraph is always smaller than the original graph. The efficiency of this sampling process depends on how many sources and targets we use during exploration. Sometimes it may occur that the subgraph we obtain has other characteristics than the underlying network graph has. If we sample a scale free network graph for example and the resulting subgraph is like a random graph, we have a certain bias occurred by the sampling process. As we normally do not know the topology of the underlying network we want to explore, those biases distort sampling processes and have to be analyzed intensively. This has already been started in several publications [2-4]. Those papers use the degree distribution as a characteristic feature of network topologies. The degree distribution is the graph obtained when plotting the number of nodes with same degree (number of connections) on the y-axis versus the number of connections on the x-axis. Degree distributions of a random graph look like a bell-shaped curve. Degree distributions of scale free graphs follow the power-law with an exponential decay. That is an important distinguishing feature between random and scale free graphs.

One special sampling bias occurs when sampling from only one or few sources [4]. Considering their degree distributions of the two topologies random and scale free are not distinguishable because both distributions look the same. Based on this observation this paper tries to acquire significant behavior differences in the progress of distributions of those two different network topologies, random graphs and scale free graphs.

The chapter *Scale Free Networks* introduces the topology of scale free graphs and where it comes from so we get a deeper insight into this topology. The chapter *Proxy for True Degree Distribution* asserts the observed redundancy as an approximation for the true degree distribution that is important to know but hard to figure out while sampling a network. In that chapter we will examine the observed redundancy of nodes, which means how often a node is traced in the process of exploration. We then compare the observed redundancy to the true degree of the node. The chapter *Approach to Distinguish between Scale Free and Random Graphs* considers one approximation to differentiate between scale free and random graphs by using the information at what time a node has been discovered in the overall process of exploration. Finally the chapter *Conclusions and Outlook* merges the two approaches and gives an outlook about what can be done next in this field of research.

Note: this thesis concentrates on complex networks in their abstract sense. Although most research is accomplished with computer simulations of explorations of the Internet topology as an example for a complex network it does not deal with computer networks in detail. The research made in this paper is applicable to any complex network because the generic node edge model the research bases on is universal adaptive.

Scale Free Networks

In this section we will have a closer look at what exactly the term scale free does describe and where to classify scale free networks together with other relevant network topologies. We then will eye several questions concerning scale free networks and extract the motivation for this thesis.

Abstract Classification

As already mentioned in the chapter *Introduction* we have a scale free topology which means that we cannot make any prediction about how many connections a node has.

When we want to have a closer look at the characteristics of network topologies the best way to explain it is to first examine the network's origin and its development, or more precisely: how the network has been assembled.

There are many different ways building a network. The later behavior of a network depends on its construction. One way of building a network is the random method as already explained in the chapter *Introduction*: we take a number of unconnected nodes. We pick two nodes per random and connect them with a probability $0 \leq p \leq 1$. By repeating this various times we obtain a random network. Every node has a predictable number of connections to other nodes. So every node's degree follows a distribution probability. The most suitable description for the degree distribution is the Poisson probability distribution given by an average degree and a variance. Its curve looks bell-shaped.

Building a scale free network is a bit more complicated than constructing a random network. We can consider the design specification as follows: like random networks we start with two connected nodes and gradually add another node to one of the existing nodes. Unlike random networks where the probability is a random number between 0 and 1, in scale free graphs it depends on the degree of the existing nodes where a new node is being connected. The more connections an existing node has, the higher is its probability for a new connection. Nodes with a high degree tend to an even higher degree whereas low degree nodes remain with a low degree. It follows the known principle "rich get richer".

The resulting network has many nodes with only one or little connections and a few nodes with a very high degree. These few but large nodes are called hubs. They play a leading part in scale free networks. If one of these nodes are attacked and taken out the whole network might split up into several smaller parts because this large hub was the only node that connected these parts to each other. But also smaller nodes may have this interconnectivity, called betweenness centrality. Figure 1 shows a small network which contains one node v with a high betweenness centrality because every shortest path from the left region to the right region coercively leads through v.

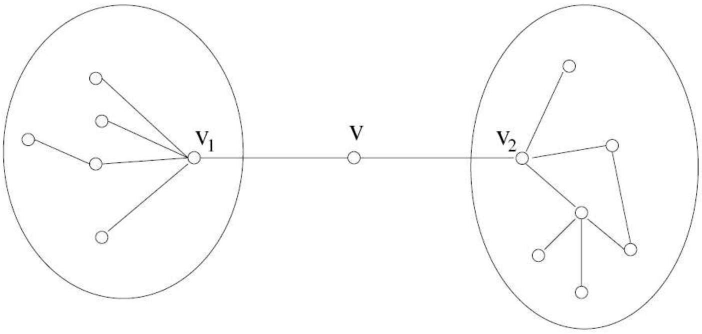

Figure 1: Network with two regions connected through only one node v. Every shortest path from the left region to the right region leads through v. This node has a high Betweenness centrality. (Adapted from ref. 7)

Dead-end nodes with only one connection have the lowest betweenness centrality. If they are removed it does not affect the functionality of the rest of the network. The higher the betweenness centrality of a node is the higher are the consequences for the network if this node is removed.

The degree distribution of scale free networks follows a power-law. The tail can be approximated with a heavy tail function which has a power-law decay:

$$P(k) \propto k^{-\gamma}.$$

It may not be mixed up with an exponential decay which falls off much steeper. To illustrate the difference between power law and exponential we have a look at the following example:

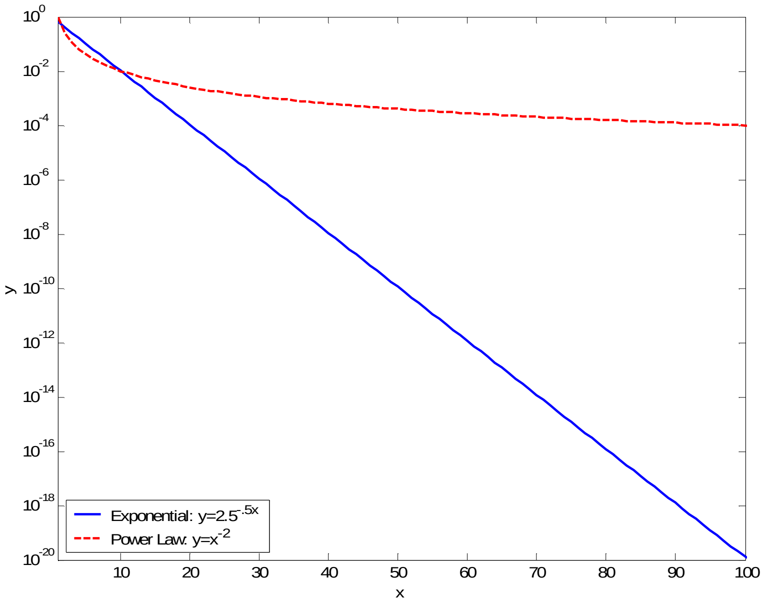

Figure 2: Exponential (blue) versus power law (red)

Figure 2 highlights the difference between exponential decay and power-law decay. The power-law falls off much smoother. The power-law decay is one characteristic feature of scale free networks.

The Internet is a popular example for a scale free network. For the Internet the exponent is $2.0 \le \gamma \le 2.5$ [5].

In Figure 5 there are examples for measurements that approximate both Poisson distribution (the black graph) and heavy-tailed power-law distribution (the yellow graph).

Open Issues

When trying to apply sampling theory to real networks like the Internet recent measurements have shown unexpected mapping characteristics. All abnormalities occurring while mapping complex networks can be reduced to the problem that when building a subgraph of the underlying network with traceroute-like probes we cannot determine the actual number of edges a node has. We do not “see” a node’s true degree when passing it with traceroute. The only edges we “see” are the incoming and the outgoing when passing a node.

One unexpected bias has been examined by Aaron Clauset and Christopher Moore in their paper [3]. They have found that when sampling complex networks from only one source a fundamental bias in observed topological features like the degree distribution occurs. They have analytically shown that sampled subgraphs of networks with Poisson-distributed degrees like random graph networks result in a power-law behavior when sampled from only one source. This observation originally was made by A. Lakhina et. al. in their publication about sampling biases in IP topology measurements [4]. They found that sampling a sparse Erdös-Rényi graph from a small number of randomly distributed sources to a larger set of randomly distributed targets provides a subgraph with a degree distribution strikingly like a power-law [4]. As described in the chapter *Introduction* this bias leads to a lack of distinguishing features between Poisson-distributed networks like random graph networks and power-law networks like scale free networks. L. Dall'Asta shows an underestimation of γ when sampling networks with the power-law distribution $P(k) \propto k^{-\gamma}$ [5]. This effect occurs due to undersampling low-degree nodes. Aaron Clauset and Christopher Moore have found that this extent of underestimation increases with the networks average degree. In their conclusion they say that confining this effect can be achieved by linearly increasing the number of sources with the average degree [3]. The question that remains is how to determine the number of needed sources because we do not know the average degree of the underlying network. That leads us to the conclusion that we have to find a way to at least distinguish between a random graph network with a Poisson distribution and a scale free network with a power-law distribution.

Sampling biases and how they occur is one of the most complex issues in this subject. There is one kind of bias we want to focus on: a sampling bias in one class of topology that lets us think that the underlying network has a very other topology [4]. A. Lakhina et al. showed that this can occur when sampling networks like the internet with a scale free topology. Especially when using only one or few sources targeting at a greater amount of targets the degree distribution appears like a heavy-tail distribution which is typical for scale free graph networks but not for random graphs. Considering the degree distribution it means with few sources there is no possibility to differ between those two topologies when sampling them. This thesis tries to go deeper into this matter and to figure out if there is a possibility to differentiate between random and scale free graphs when sampling them. That might lead then to an approach eliminating the described sampling biases.

Proxy for True Degree Distribution

When sampling the Internet on router level with a traceroute-like program we only get a very little fraction of nodes from the whole network. Each traceroute delivers an amount of detected nodes and each node has an amount of connections to other nodes. But when a traceroute passes one node we only see two connections with that traceroute: the incoming and the outgoing. What we can't see with traceroute is the true degree of that node. An exception may be sampling methods on AS level: there the true degrees of nodes actually can be seen because the BGP protocol used as sampling protocol provides information about the number of connections a router has [13]. The rocketfuel project for example avails BGP routing tables to improve scans of ISP topologies on router level. In their paper N. Spring et. al. showed that the number of traces could be reduced by three orders of magnitude without a considerable loss of accuracy unlike brute-force, all-to-all scans [9].

Another exception of not seeing a node's true degree while sampling the network might be the WWW, as the number of outgoing links of a website easily can be determined by scanning the HTML code.

But native traceroutes and many other network exploration tools for other networks do not provide this desirable information. So we now try to find a proxy for the true degree in networks when sampling them.

One idea is to consider the number of discoveries of one node as a proxy for its degree. As already mentioned it occurs that while tracerouting a network several nodes may be traced multiple times. The assumption is: the bigger the degree of a node is the more traceroutes pass through it.

Simulation Procedure

To gather information that can be used to determine whether or not the assumption is true we need to set up a whole chain of events:

1. Network generation with a network generator tool
2. Import of the generated network for further computations
3. Deployment of sources and targets
4. Calculation of shortest paths for each source
5. Simulation of traceroutes through the imported network
6. Exportation of results for further analysis with Matlab
7. Analysis and plotting with Matlab

Most network generator tools store their generated networks into ASCII files with the L1-L2 format. Files with this format consist of *m* lines, where *m* is the number of links the network has. Each line in the file consists of two node ids between which the link exists. The node ids are sorted ascending.

The networks that were explored for this thesis were generated with PFPModel (a scale free network generator) and a random graph generator, both provided by Dr. Shi Zhou [6]. Other network generators such as brite [10] or INET [11] came into consideration but lead to some difficulties considering compiling the software. PFPModel and the random graph generator provided by Dr. Shi Zhou were ready-to-start executables and complied with requirements. To what extend the choice of these generators influences the outcomes of the simulations is not determined as simulations with networks of other generators have not been executed yet.

Steps 2-6 of the simulation procedure chain are covered by the self written simulation software [8] which we are going into in the following paragraphs.

The program is written in C++. I have chosen that language because we both are close to the hardware so the speed of computations is quite fair and C++ provides object oriented structures and containers that help out with memory management and covers a lot of features such as push() and pop() for queues for example.

Objects like networks, nodes or shortest paths have been mapped to classes. Each node of a network for example is an instance of the class *Node* and is stored in a container that is part of an instance of the class *Network*. For each simulation run the program instantiates two instances of the class *Network*. The first contains the underlying network that is sampled. It is built from the L1-L2 file. The second contains the subgraph that is filled during the exploration process.

The program reads a L1-L2 network from an ASCII file and builds the corresponding network in a class oriented structure. After building up the corresponding network in memory the program deploys sources and targets. The numbers of sources and targets can be determined by a shell parameter. We also can influence the kind of deployment: random, lowest degree or highest degree deploy. The deployment mode used for the simulations has always been set to random.

After deploying the sources and targets the program uses the shortest path algorithm adapted from ref. 3 and computes shortest paths for every source we want to simulate traceroutes from. Here is the algorithm in pseudocode as printed in ref. 3:

```
while there are pending vertices:
    choose a pending vertex v
    label v reached
    for every unknown neighbor u of v;
        label u pending.
```

To start this algorithm one node in the network is labelled pending. It then works as long as every other node in the network has been chosen. When a node is chosen the distance to the origin is stored. This creates a kind of "routing table" for every node.

After the shortest path algorithm has calculated all relevant shortest paths the software then calculates the traceroutes as follows: the first traceroute is simulated from the first source to the first target. The second traceroute is calculated from the second source to the second target and so on. While simulating the program memorizes every newly discovered node and the time (in traceroute simulation steps) it has been discovered. Every time a traceroute passes a node, the program increments a counter which counts the number of multiple discoveries. At the end of the traceroute simulations the program stores the relevant data node id, true degree, number of multiple discoveries and time of first discovery in an output file.

In the simulations one source was randomly deployed and all[1] other nodes in the network were targets. The results that have been stored in a file were evaluated with Matlab. To centralize all information Matlab memorizes for each degree all redundancies of nodes with that degree. Then we are able to plot the observed redundancy versus the true degree. With that plotted curve we can visually evaluate whether or not the approximation assumption can be confirmed and is applicable for further network explorations.

As introduced in [5] we can consider the level of sampling of networks with

$$\varepsilon = \frac{N_S N_T}{N} = \rho_T N_S .$$

N_S and N_T represent the number of sources and targets deployed in the network. Because we consider explorations with only one source and much more targets we declare ρ_T as density. In this special case ρ_T is ≈ 1: we have one source and the other nodes of the network are targets.

[1] The scale free networks consist of 100,100 nodes but only 99,999 of them were targets, so „all" is only correct for random graph networks, and it were approximately „all" for scale free with an error $\leq 1\%$.

On the x-axis we plot the true degree. On the y-axis we plot the median of all redundancies of the nodes with same degree that have been traced during one simulation run. I decided to use the median but not a mean algorithm because the median algorithm eliminates interfering outliers. There was an alternative to use Matlab's boxplot function. But as we need both axes logarithmic and Matlab scales the width of the boxes to the x-axis the boxes appear variably wide. To me the boxplots are inapplicable.

Although the analyses are based on median plots I provided one mean plot and one boxplot to exemplarily show the differences of the evaluations.

Results

Figure 3 provides a plot of the first result set. The median of observed redundancies is plotted versus the true degree. As we see in that figure we have a monotone relationship between the observed redundancy of nodes and their true degree. It is no linearity because both x-axis and y-axis are logarithmic. Linearity in log-log plots denotes an exponential relationship between x and y. Hence the relation between observed redundancy and true degree follows a power-law. We could not have seen that relationship without plotting the curves into a log-log coordinate system.

Basically the statement Figure 3 expresses is: larger nodes are traced more often than smaller nodes. This may lead to the conclusion we can use the observed redundancy as a proxy for the true degree distribution of a sampled network.

Note: neither do we develop the mathematical equation describing this dependency nor is that relation relevant for essentially exploring the basics of approximation this chapter aims.

What we have to keep in mind is that the data the graph is plotted from is taken from a complete exploration, which means that every node was explored. When sampling real networks we do not have the opportunity to explore them completely. So we have to use this approximation carefully.

In order to ensure that these measurements are not only coincidence I executed all simulations with five independently generated networks on five different machines. Visually they look similar. We do not want to have all five measurements of the networks averaged because in real life we have only one Internet and not the average of five Internets. So we consider only one network as an instance. To prove measurements are no coincidence there is plotted a second set of network exploration simulation curves in Figure 4. They indeed look very similar to those in Figure 3, except for one outlier of the blue curve in Figure 4.

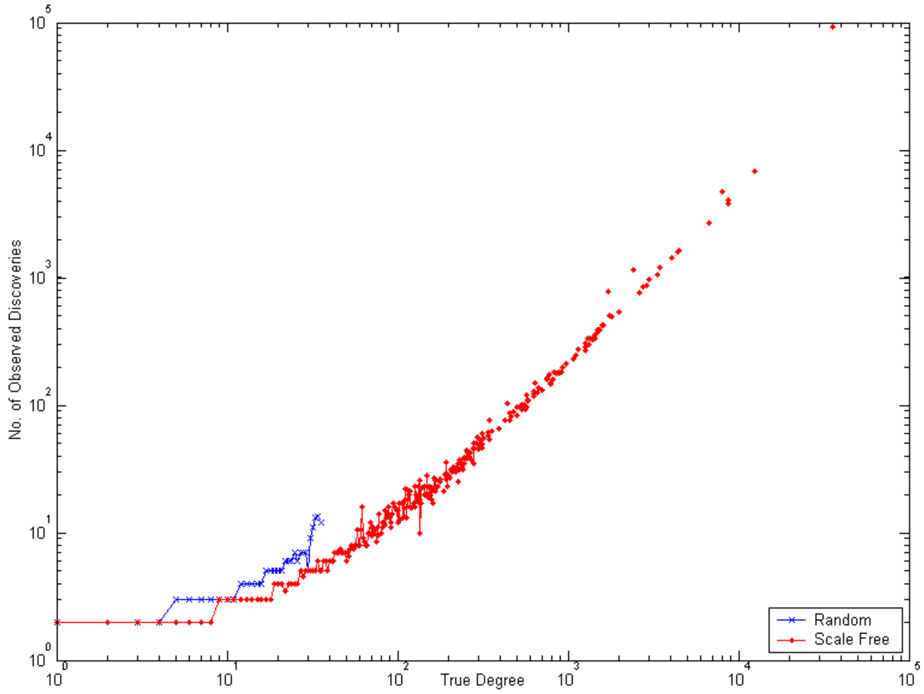

Figure 3: Median of number of multiple discoveries of nodes vs. their true degree of two explored networks with 100,000 (random) and 100,100 (scale free) nodes respectively, the average degrees are 6.0 (random) and 5.4 (scale free), network explored from one source to every node using shortest path

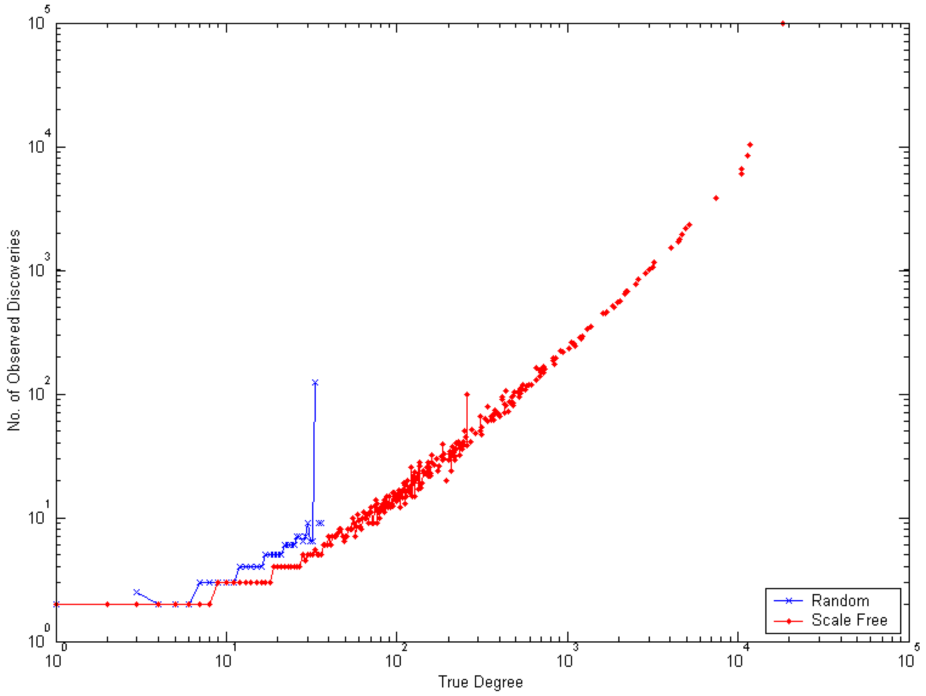

Figure 4: Another test series with independently generated underlying networks with criteria as described for Figure 3

The curves in Figure 3 and Figure 4 are stepped. This phenomenon occurs due to the median average determination which always results in multiples of 0.5. Figure 8 provides a plot of the same data Figure 3 is plotted from but with mean average determination. Hence the curves in Figure 8 do not appear stepped but more even.

Figure 3 shows the number of multiple discoveries of the two network topologies random and scale free. What might be interesting to know in addition is where the maximum of the Poisson distribution of the random graph lies and how the scale free degree distribution looks like. Perhaps there might be any kind of coherence between the multiple discoveries and the true degree distribution. Therefore we just superpose their true degree distribution.

Figure 5 shows the plots provided by the same traceroute simulation for the data plotted in Figure 3, but now superposed with the corresponding true degree distributions of their underlying networks.

Considering Figure 5 the coherence between the particular curves of both random (blue and black) and scale free (red and yellow) networks can be verified. The random curves (blue and black) only vary in lower degree ranges from 1 to 36, whereas the two scale free curves (red and yellow) reach up to over 10,000. The black curve looks Poissonian, excepting the degree 1. The yellow curve shows typical power-law behavior. We also see the power-law decay, a slowly decreasing curve that ranges to over 10,000, just like a typical heavy-tail.

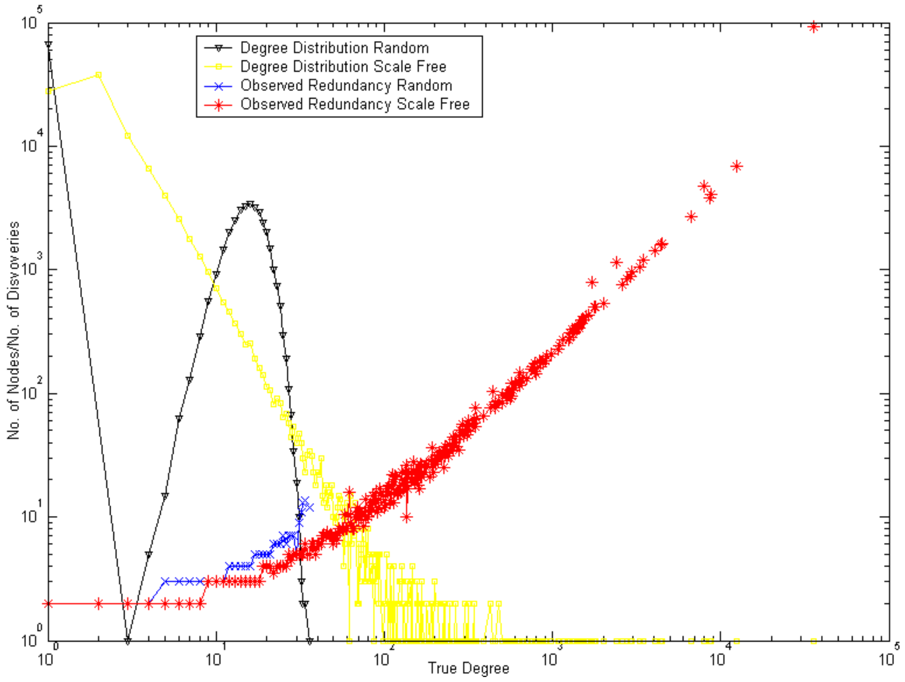

Figure 5: The same as in Figure 3, now superposed with corresponding degree distributions of the underlying networks

As the observed redundancy seems to be applicable as a proxy for the true degree we may plot their distributions and look how close they are together. Although we know it is no linear proxy we want to have a look at the approximated curves anyway in order to compare them to the original distribution and in order to extract potential hints.

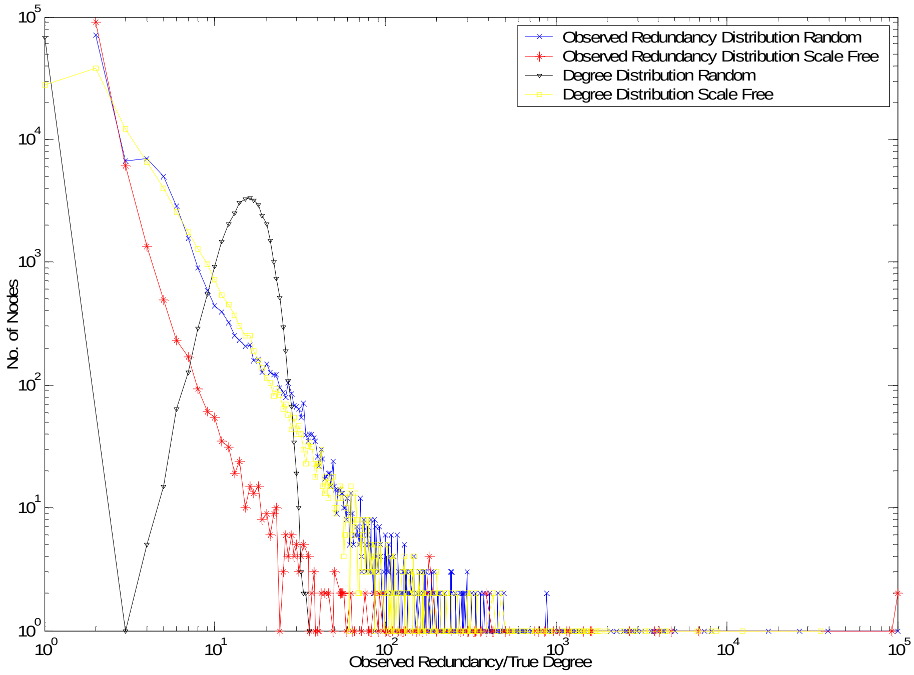

Figure 6: Distributions of both observed redundancy and true degree of random and scale free networks respectively

Ideally the blue curve in Figure 6 should look like the black one and the red curve should be like the yellow one. This instance would be if the proxy was exactly linear. But as it is not the observed redundancy curves are biased.

The scale free approximation (red) looks quite close to its original (yellow), especially the heavy-tail is perfectly visible. And also the random network approximation shows at least a hint of a maximum it actually should present like the true degree distribution does. But here we also obtain a heavy-tail for the random approximation (blue), similar to the sampling biases already described in other publications [3, 4].

As expected the random degree distribution in Figure 5 (black curve) looks very Poisson-like. What mentioned in ref. 3 and 4 was that the degree distribution of a sampled random graph may look like a heavy-tailed power-law as it occurs in scale free networks. But to observe this behavior it is called for generating a random graph with a much higher average degree like shown in ref. 4. Therefore we generated a random network with 800 nodes and 160,000 links, which equals to an average degree of 400.

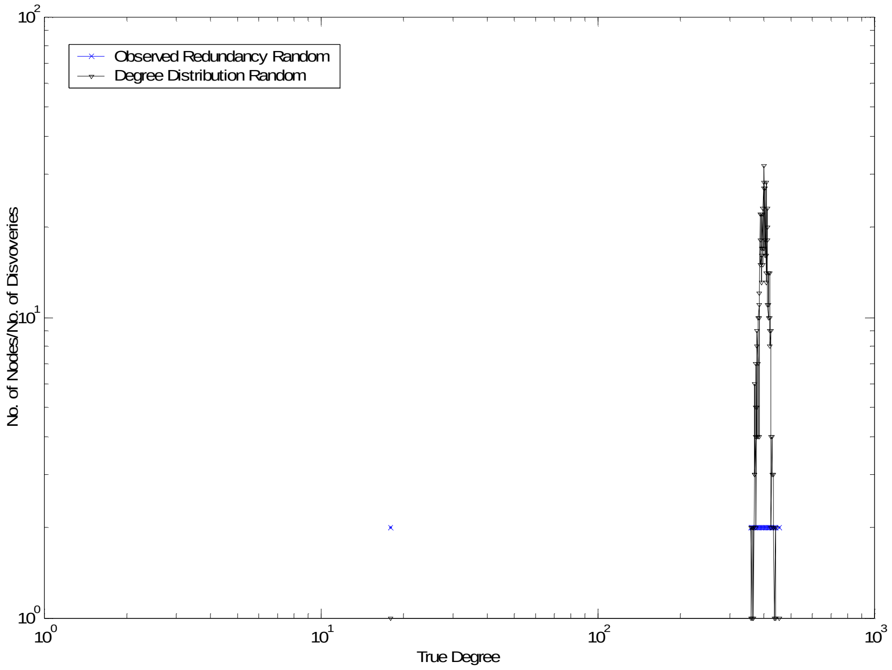

Figure 7: number of multiple discoveries (blue) and degree distribution (black) in a random network with an average degree of 400

The degree distribution shows all nodes degrees range from 350...450 with a maximum at 400. In this traceroute simulation we can see that the observed redundancy is quite constant. Beside a few outcasts (discarded by the median algorithm) all nodes have been traced two times as seen in Figure 7, blue curve.

Obviously the approximation *observed redundancy – true degree* cannot be made for high average degree networks.

Figure 8 provides the same data evaluation of exploration simulations like in Figure 3 except its average determination is now mean instead of using the median. This leads to smoother curves that do not appear as stepped as they appeared with the median average determination. And we now see outliers that have been eliminated by the median.Obviously these must be the source nodes factored into the average determination as a regular node observed 99,999 times. Several other nodes with the same degree but not being traced that much lead to an average observed redundancy much higher than their median pendant. Due to the average determination both peaks in the curves of course do not reach up to 99,999.

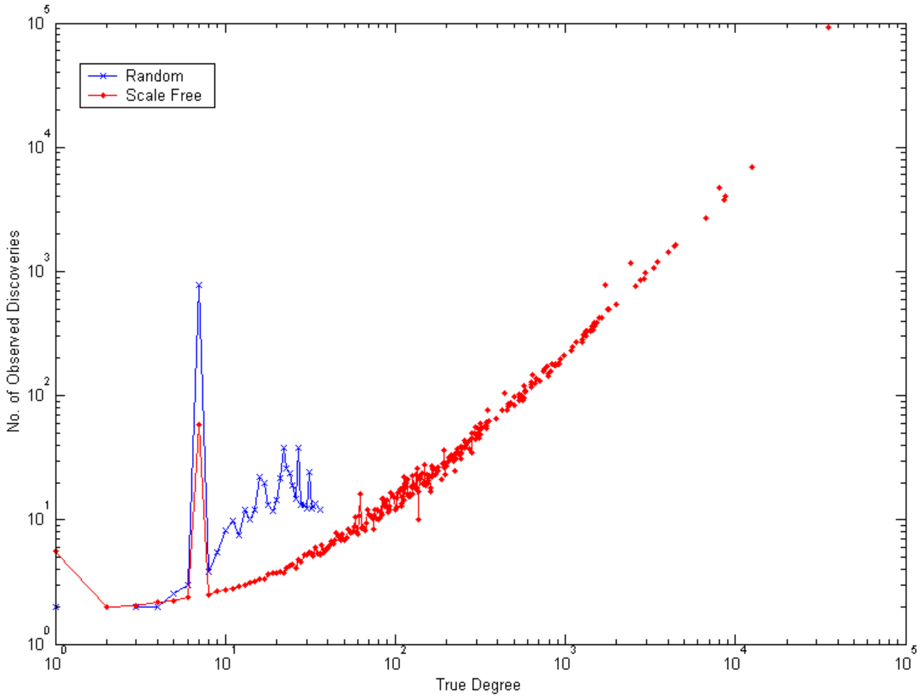

Figure 8: Mean of number of multiple discoveries of nodes vs. their true degree of two explored networks with 100,000 (random) and 100,100 (scale free) nodes respectively, the average degrees are 6.0 (random) and 5.4 (scale free), network explored from one source to every node using shortest path

In order to see the dispersion of the whole data I generated a Matlab boxplot of the observed redundancy of the scale free test series.

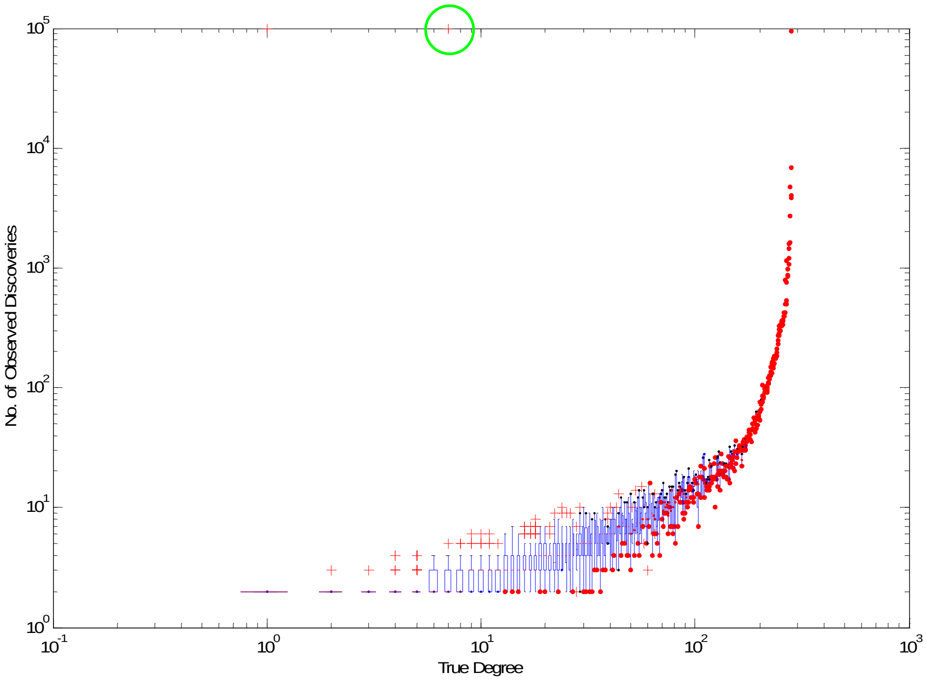

Figure 9: Boxplot of number of multiple discoveries of the scale free test series. The green circle marks the outlier that we also can see in Figure 8.

In Figure 9 we see the very high redundancy outlier that leads to the outlier in Figure 8, highlighted by the green circle. As the x-axis is logarithmic the boxes are not constantly wide. In my opinion a boxplot with a logarithmic x-axis looks improper. Hence we abdicate considering more boxplots.

Approach to Distinguish between Scale Free and Random Graphs

As already mentioned in the chapter *Introduction* ref. 3 has revealed that network explorations from only one source in random graph networks with a high average degree lead to a sampling bias that lets the degree distribution of the sampled subgraph look heavy-tailed, like a distribution from a scale free network graph. What we want do is to find a way to avoid this bias, or, if we cannot, to although distinguish between the two sampled subgraphs without knowing what kind of topology we sampled.

M. Barthélemy et. al. studied the spread of diseases in scale free networks and focused on the predisposition of large nodes, respectively hubs, which are infected quite early in the time evolution of epidemic outbreaks [12]. They found that the predisposition of hubs also depends on the connectivity around the initial seed and the distance of the seed to the nearest hubs.

The idea for the distinction is to have a look at the development of the subgraph while sampling the original graph, especially to determine the particular times when nodes are discovered first and then to have a look at their true degree.

Hence the particular times of discovery of each discovered node has been memorized while simulating traceroute explorations as described in the former chapter.

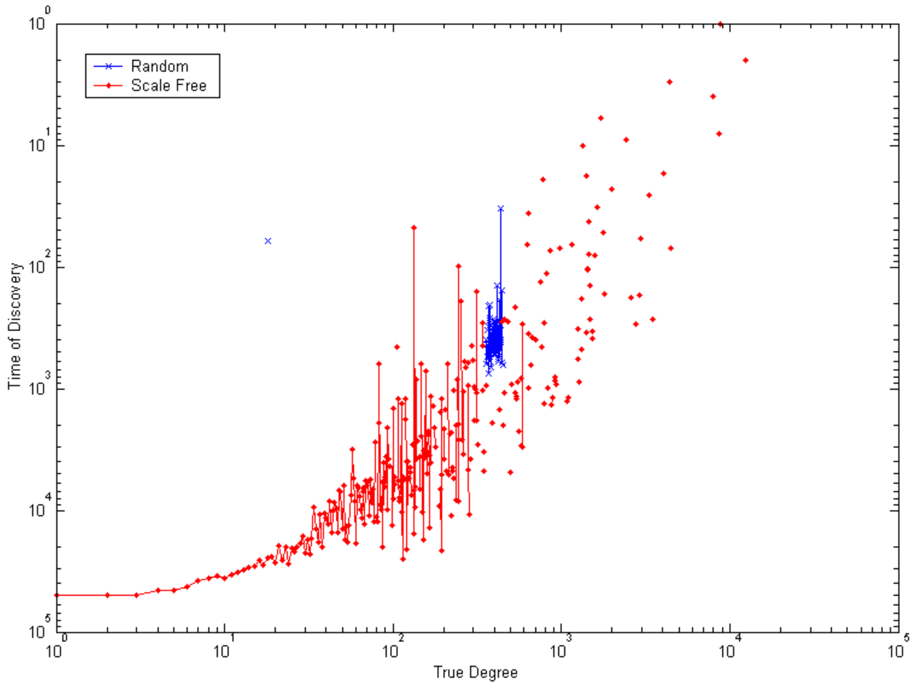

Figure 10: Time of Discovery vs. True Degree, measured within the same simulation as in the former chapter

Figure 10 shows the median time of discovery of nodes plotted versus their true degree. It shows that in scale free networks (red curve) large nodes with a high true degree are discovered earlier than nodes with a small degree. But for the random graph network (blue curve) there can be made no assertion at all because the curve looks very noisy.

The wider range of the red scale free curve on the y axis is because it has much more nodes to explore. The scale free network in this simulation consists of 100,100 nodes, the random network of 800. This is due to the random network generator: it did not deliver considerably larger graphs with high average degree. So the sampling process of the random network already ends after 799 traces, whereas the scale free graph was tracerouted 99,999 times. The red curve already ends at 50,000 due to the median algorithm.

Focusing on the very sharp and clean left tail of the red curve in Figure 10 we definitely can distinguish between the two graphs. But to obtain this tail the scale free network has to be sampled with a very high density in order to reach this late point of progress: the lower we get on the y-axis the farer the sampling progress is.

In the beginning of sampling both random and scale free graphs (upper y-axis area) we only obtain noise.

Conclusions and Outlook

As we have seen in chapter *Proxy for True Degree Distribution* we may under certain circumstances use the observed redundancy as a proxy for the true degree of networks. In chapter *Approach to Distinguish between Scale Free and Random Graphs* we have detected a light difference between random and scale free networks while exploring them. Now we can merge both approaches and use the first subject *observed redundancy* to reason in the second issue that when observing early detected nodes over and over again we can assume with high probability that we have a large node.

Figure 3 provides an approximation for the true degree by the observed redundancies of multiply explored nodes. As mentioned in the chapter *Proxy for True Degree Distribution* we do not know yet a formula describing this approximation. It also depends on many factors like number of nodes, average degree and number of sources/targets. For further explorations into this field of study the influence of these factors on the outcome of the approximation should be examined. The second aspect to go into is the mathematical coherence of the observed redundancy and the true degree. As we did not develop any equation approach for this approximation this could be interesting to analyze.

What also would be very interesting and to be done next was to simulate network explorations considering their hierarchy versus true degree with larger random graphs with high average degree. In this thesis we only have considered random graph networks with average degree 400 but only 800 nodes, chiefly because the random graph generator did not provide networks with a considerably higher average degree. So that subject only has been touched. Explorations of random networks with a high average degree and a large number of nodes (10^6 and more) would be desirable to analyze biases more effectively.

In this paper we mostly made qualitative statements based on numerical experiments and their observations. There still is a highly lack of quantitative propositions concluding from the numerical experiments. This thesis provides well-summarized information to develop new ideas and gives the groundwork for further explorations into this field of research.

References

[1] S. H. Strogatz, "Exploring complex networks", *Nature* **410**, 268-276 (2001)

[2] W. Willinger, R. Govindan, S. Jamin, V. Paxson, and S. Shenker, "Scaling phenomena in the Internet: Critically examining criticality", *Proc. Natl. Acad. Sci USA* 99 2573-2580, (2002)

[3] A. Clauset, C. Moore, "Accuracy and Scaling Phenomena in Internet Mapping", *Physical Review Letters* **94**, 018701 1-4 (2005)

[4] A. Lakhina, J. W. Byers, M. Crovella, P. Xie, "Sampling Biases in IP Topology Measurements", in *Proceedings of IEEE INFOCOM* San Francisco, CA, 2003

[5] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, A. Vespignani, "A statistical approach to the traceroute-like exploration of networks: theory and simulations", *cond-mat*/0406404 (Jun 2004)

[6] Dr. Shi Zhou, University College London, Adastral Park Postgradual Research Campus: s.zhou@adastral.ucl.ac.uk

[7] M. Barthélemy, "Betweenness Centrality in Large Complex Networks", *Eur. Phys. J. B* **38**, 163–168 (2004)

[8] Routing Simulation Software "SimTrace", contact dankbar@fh-muenster.de

[9] N. Spring, R. Mahajan, D. Wetherall, „Measuring ISP Topologies with Rocketfuel", in *Proceedings of ACM SIGCOMM*, August 2002

[10] brite: Boston university Representative Internet Topology gEnerator: http://www.cs.bu.edu/brite/

[11] INET: http://topology.eecs.umich.edu/inet/

[12] M. Barthélemy, A. Barrat, R. Pastor-Satorras, A. Vespignani, "Velocity and Hierarchical Spread of Epidemic Outbreaks in Scale-Free Networks", *Physical Review Letters* **92**, 178701 1-4 (2004)

[13] Information provided in private notification of Arnold Nipper, DE-CIX Management GmbH (German Internet Exchange)