DEGREE PROJECT IN TECHNOLOGY, FIRST CYCLE, 15 CREDITS



**KTH ROYAL INSTITUTE OF TECHNOLOGY**

**SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION**

*STOCKHOLM*, *SWEDEN 2016*

Examining the structure of the KTH web

**ALEXANDER JANSON ERIK SNICKARE**





**Undersökning av KTH webbens struktur**

#### ALEXANDER JANSON ERIK SNICKARE

Degree Project in Computer Science, DD143X Supervisor: Arvind Kumar Examiner: Örjan Ekeberg

CSC,KTH 2016-05-11

# Abstract

This thesis studies the characteristics of the network structure extracted from the public KTH web. The network structure was extracted with a *crawler* and consisted of 671,013 nodes and 23,515,683 links. The system is studied by applying statistical concepts from network science such as *degree distribution* and *average path* to reveal the characteristics of the network. The aim of the statistical analysis is to explore the *robustness* of the network and to answer if the network is *scale-free*. The thesis will examine the results from the study and compare the results to previous similar research.

The results indicate that there might have been a change in regards to the network structure of websites since the last major research was done on the subject, likely caused by changes in web design. However, the results still indicate characteristics typical for a scale-free network. Due to irregularities in the crawler results may be slightly unreliable.

3

# Sammanfattning

Detta kandidatexamensarbete undersöker de egenskaper i nätverksstrukturen för det allmänna KTH nätet. Denna struktur extraherades med en spindel och bestod av 671,013 noder och 23,515,583 länkar. Systemet studerades genom att applicera statistiska koncept från nätverksteori såsom gradfördelning och genomsnittlig väg för att visa nätverkets egenskaper. Målet med den statistiska undersökningen är utforska nätverkets robusthet och besvara frågan om nätverket är så kallad *scale-free*. Studien jämför även resultaten med resultat från tidigare studier med liknande undersökningar.

Resultaten indikerar att det kan ha blivit en förändring i avseende på nätverkets struktur sedan den tidigare omfattande studien genomfördes, förmodligen orsakat av ändringar i webdesign. Trots det så indikerar resultaten att nätverket fortfarande har egenskaper typiska för ett *scale-free* nätverk. På grund av oregelbundenhet i spindeln så kan resultaten vara smått opålitliga.

Contents

[Abstract 3](#_bookmark0)

[Sammanfattning 4](#_bookmark1)

1. [Introduction 6](#_bookmark2)
   1. [Problem statement 7](#_bookmark3)
   2. [Scope 7](#_bookmark4)
   3. [Thesis overview 7](#_bookmark5)
2. [Background 8](#_bookmark6)
   1. [Graphs 8](#_bookmark7)
   2. [Networks 9](#_bookmark8)
      1. [The World Wide Web 9](#_bookmark9)
      2. [Complex random networks 9](#_bookmark10)
      3. [Small-world networks 10](#_bookmark11)
      4. [Scale-free networks 10](#_bookmark12)
      5. [Duplication-divergence model 10](#_bookmark13)
      6. [Robustness of networks 11](#_bookmark14)
   3. [Crawlers 12](#_bookmark15)
      1. [Properties of a crawler 12](#_bookmark16)
3. [Methods 14](#_bookmark17)
   1. [Extraction of web pages 14](#_bookmark18)
   2. [Investigated graph properties 14](#_bookmark19)
      1. [Degree distribution 14](#_bookmark20)
      2. [Average path length 15](#_bookmark21)
   3. [Definition of robustness 16](#_bookmark22)
4. [Results 17](#_bookmark23)
   1. [Degree distribution 17](#_bookmark24)
   2. [Efficient Average path approximation algorithm by exploiting degree distribution 19](#_bookmark25)
   3. [Average path length of the KTH web 20](#_bookmark26)
   4. [Robustness of the network 21](#_bookmark27)
5. [Discussion 23](#_bookmark28)
   1. [Results analysis 23](#_bookmark29)
   2. [Limitations 24](#_bookmark30)
   3. [Future studies 24](#_bookmark31)
   4. [Conclusions 24](#_bookmark32)

[References 25](#_bookmark33)

# Introduction

A network is a set of items and their connections to one another [1]. Networks exist practically everywhere in all shapes and sizes and can include everything from informational networks, like the Internet where routers and cables form a network, to biological networks which contains molecules, neurons etc [2]. Together networks form the foundation for network theory.

One major breakthrough in network theory came at the end of the 1990’s when large amounts of data sets became available in the form of the World Wide Web [3]. The World Wide Web (WWW) refers to all the Web pages accessible on the Internet and the WWW forms a network by connecting the web pages with so called URLs [2]. In 1999 it was discovered that the WWW was a scale-free network where the vast majority of nodes had few incoming and outgoing URLs [4]. Furthermore, it was also observed that, in spite of the networks large size, most web pages were reachable by following only a few URLs [4]. This is known as a small-world property [4]. In the following years more networks, like the Internet [5], seemed to follow the scale-free model. Even non-human constructed networks, such as the cellular metabolic network [6], seemed to follow the model. To understand the growth of such complex systems, mechanisms like preferential attachment [7] emerged, where webpages with already many connections are likely to get more connections. Another proposed growth model is duplication-divergence [8] that mainly relies on copying already existing parts of the network.

Robustness is another highly examined network property and describes how networks behave when connections are removed from them [9][10] . This is an important functionality to bear in mind when providing error tolerant systems, whether someone is trying sabotage it or just some random failure within it [9]. In essence the understanding of one network can lead to the understanding of several distinct networks. This in turn provides tools to help and shape already existing systems to become more robust and fit for their purpose.

Today the WWW is thousands of times larger than it was in the 1990’s and contains over one trillion pages [11]. With such a large increase of websites, and with how web development might have advanced over the past decade, it would seem peculiar if it maintained its initial properties. This study will reexamine the properties of the WWW with the aim to see if anything significant has changed to its structure in terms of degree distribution and robustness.

## Problem statement

This thesis will focus on extracting and studying the network structure of the website kth.se. Studying the network will be done in terms of robustness and dynamic properties of the graph. The study intends to answer the following questions:

* Does the network have a power-law degree distribution and a small-world property?
* How robust is the kth.se network?

## Scope

The network that will be analyzed for this thesis is the network of the public (non-user logged in) kth.se web. The extraction will be done by traversing links that contain “kth.se/” and all other links will be dismissed. Kth.se was chosen both due to it being our university affiliated website, but also due to the network examined previously in 1999 by Albert, Jeong and Barabás being a university website [4]. The thesis will not take into account the differences between the network of an authorized user and an unauthorized user. The structure of the web will be considered constant during the time which the thesis is done and no changes done to the real kth.se website will be taken into account for. Explored graph properties of the network will be chosen with consideration to how well they represent the network’s characteristics and robustness.

The chosen algorithms to calculate the graph properties will be done based on the execution time and memory usage. Due to the large size of the network, limited timeframe and computation power in combination with the algorithms complexity, the values for the average path will be estimated instead of calculated exactly. Another limitation will be the lack of customizability for the crawler to make it extract a fully correct representation of the network.

## Thesis overview

Section two describes the background and the concept of the most important subjects of this thesis. Divided into three sections section two first describes graphs while the second part describes networks and the third part describes crawlers and their properties. The thesis then moves to section three which describes how the thesis was done and how the results were gathered. Section four describes the result of the gathered information about the system and what was noteworthy about the results. The last section concludes the report with in-depth discussion about the results presented in section four and a summary of the report.

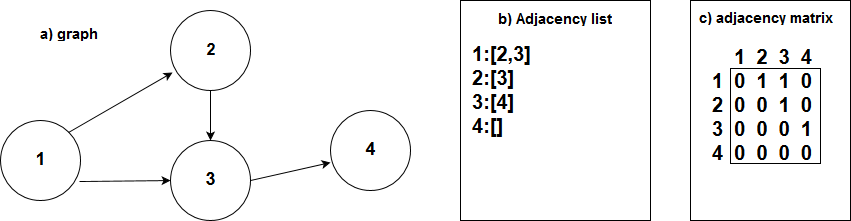
# Background

In these sections we will briefly present graphs, networks and their structure, robustness of networks and crawlers along with their properties.

## Graphs

A graph is a set of vertices that are connected by edges [12]. If two vertices are connected by an edge they are said to be adjacent. Adjacency is mainly represented as the data structures *adjacency list* and *adjacency matrix*. Adjacency lists in the simplest form consists of a list of all vertices where each vertex has a list of only its adjacent vertices [13]. However, in an adjacency matrix the list of each vertex contains all vertices in the same order as the vertices list, containing whether the vertices are adjacent or not [12]. A path is a set of edges that connects one vertex to another [13]. A set of vertices that have paths between themselves is said to be a component if they are unreachable for any other vertices in the graph.

A graph can have either directed or undirected edges [12]. Undirected edges means that if the vertex *u* is adjacent to the vertex *v* then *v* is also adjacent to *u* . On the other hand, in directed graphs *u* being adjacent to *v* does not imply that *v* is adjacent to *u* . If an edge in an directed graph goes from *u* to *v* then the edge is *u*'s outgoing edge and *v*'s incoming edge. In undirected graphs the edge is both incoming and outgoing for *u* and *v*. The number of adjacent vertices to a vertex is known as its degree [12]. The incoming degree describes the number of incoming edges to a vertex and the outgoing degree describes the number of outgoing edges from a vertex.

Edges of a graph may also have a number attached to it known as weight [12]. Graphs with such weighted edges are referred to as weighted graphs. Weighted graphs with only positive values are specified as unsigned graphs, whereas graphs with mixed positive and negative weights are known as signed graphs [14].

**Figure 2.1:** a) A directed unweighted graph and b) its adjacency list representation and c) its adjacency matrix representation

## Networks

A network is described as a set of items with relations between them [1]. This structure allows networks to be represented as a graph where the items are vertices and the relations are edges. A vertex with relatively many edges is referred to as a hub [2]. The hubs are important to keep the network fully connected.

Networks can be categorized in several ways. One way to categorize networks is by their degree distribution, by finding some general patterns which each vertex follows. Another way to categorize networks is by how they grow, as in how does new vertices connect to a network as they are added. Lastly, there are various features to the networks which can characterize them, like the average path of a graph.

### The World Wide Web

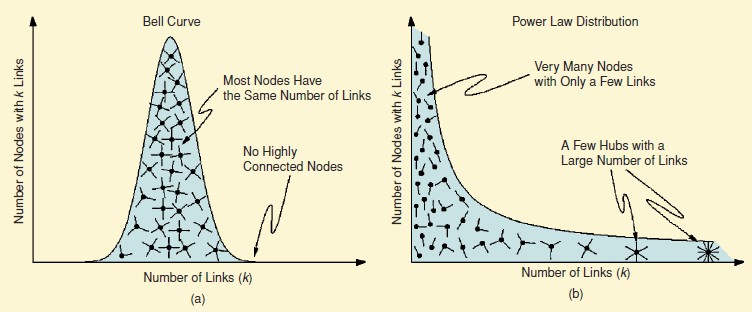
The World Wide Web (WWW) refers to all the Web pages accessible on the Internet [2]. Web pages can contain uniform resource locators (URLs) that can be used to access another web page. Therefore the WWW network can be represented as a directed graph with the URLs as relations and web pages as vertices.

The WWW contains over a trillion web pages [11], in contrast the average path from one web page to another being around 19 [1]. When large networks have small paths is often referred to as a small world property and this property is also shared in other networks such as social networks [15].

### Complex random networks

One of the simplest models of a random network was described by Erdos and Rényi [2]. In such ER-type random networks, nodes connect to each other with a fixed probability. Erdos and Rényi presented graphs with a set amount of nodes and then randomly connected nodes to one another. When they examined the results every node had roughly the same amount of edges. In statistical terms when all vertices have a high probability of having the same amount of edges the graph is said to have a “Poisson distribution” [2].

Some networks that are found in natural systems differ from the ER-type random networks. For instance, in a study performed by Barabási, Albert and Jeong using a crawler to map the number of incoming URLs and outgoing URLs on 325,729 web pages the results showed a power-law distribution [4]. In this distribution the vast majority of vertices have a low amount of edges which get exponentially fewer the more edges they have. It was later shown that more kinds of networks followed power-law distribution, for example the routers and their connections on the Internet [5].



**Figure 2.2.2:** Image illustrating a) poisson distribution and b) power-law distribution. *Source: IEEE Control Systems Magazine* [2] *figure 1*.

### Small-world networks

Small-world networks, as indicated by the name, are networks that are defined by their high amount of clustering, as in the vertices and their connections grouping up [16]. Another property of small-world networks is that the distance between vertices is fairly low and scales by L= Log(N) where *L* is the distance and *N* is the number of vertices.

### Scale-free networks

Networks that have a power-law distribution are also called scale-free networks [7]. In a model presented by Barabási and Albert these networks also follow two general mechanisms known as *growth* and *preferential attachment*. The authors of the model showed via simulations testing each independently that both were necessary to create a power-law distribution of the network [17].

* They *grow*. All the time new vertices are being added and connected to the network. What this means in practice is that the networks do not remain static.
* They have *preferential attachment*. What this means is that more popular vertices are more likely to have a connection to the new vertices than vertices with very few connections.

### Duplication-divergence model

The duplication-divergence model plays a key role in the growth of biological networks, but is also claimed to affect the addition of new web pages and their connections [8]. Like the scale-free networks duplication-divergence follows two mechanisms, *duplication* and *divergence*.

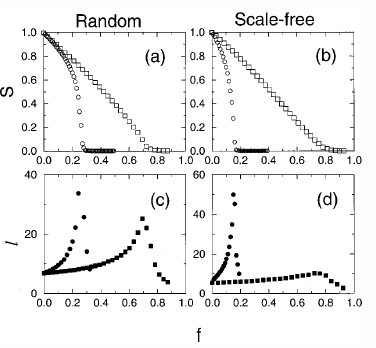
* They *duplicate.* A random vertex is selected to duplicate itself, meaning that a new vertex is created that has edges to the same vertices as the duplicated vertex.
* They *diverge*. At the stage of duplication, each edge is given a set probability σ to be kept in the duplicated vertex and a (1- σ) chance of removal. If all edges are removed the vertex is removed as well.

### Robustness of networks

The robustness of a network is usually connected to the error tolerance of the network. Error tolerance, in the context of networks, describes how the removal of nodes affects the overall connectedness of the network.

One way of measuring the robustness of a network is with percolation [2]. Percolation theory is a set of tools used to understand complex networks [7]. One way to add percolation to a graph is through the random removal of nodes and observe the critical point where the graph is divided into several components [2]. It was shown in [9] that scale-free networks have a higher level of robustness to these random node removal than completely random networks. The main reasons were that scale-free networks generally had a higher chance of removing the node with few connections due to the power-law distribution.

However, the hub structure that makes these networks robust to random edge attacks, also makes them vulnerable to targeted edge attacks [9]. For example the removal of hubs can quickly fragment the scale-free network into smaller networks, whereas in a ER-type random network this would not be such a high vulnerability.



**Figure 2.2.6:** Results of robustness of random and scale-free networks in a previous study. The relative size of largest cluster (a), (b) and average path length in (c), (d) for different networks. Network sizes are 10,000 nodes and average degree 4. ☐, random node removal,

◯ targeted attack. *Source: Rev. Mod. Phys*. [7] Fig 32

Targeted attacks against networks governed by duplication-divergence mechanisms showed that these kinds of networks had less robustness than scale-free networks [10]. This behavior is closer to observations done on real websites [10]. Furthermore, the higher the average degree or probability σ was observed to be more robust to targeted attacks.

## Crawlers

A web crawler is a program that systematically parses or browses the World Wide Web to retrieve information [18].

A web crawler parses the World Wide Web by first visiting a pre-determined set of URL’s, called seeds. By extracting the hyperlinks found in the pages of the seeds the crawler gains a new set of URL’s to visit, called the crawl frontier [18].

Web indexing is in turn used to improve searching the web, using methods such as Google’s PageRank. Google’s PageRank ranks pages based on how many times it is linked to from other web pages. The algorithm also takes the quality of these links into account, where a link from a page with a higher PageRank is regarded as higher quality [19].

The PageRank of a set of web pages can be calculated either from an adjacency matrix or from a graph. The adjacency matrix and the graph is constructed from information gathered by a crawler [20][21].

Referrals from search engines account for 41% of the external traffic a domain receives [22].

There are different ways to sample data from the web. Since the World Wide Web today is so vast and complex it would be impossible to visit all the web pages and to at the last web page still have relevant information about the first web page. Therefore the number of web pages on the internet could be regarded as practically infinite. Because of this there has to be some restrictions on a crawler as to what pages should be visited and what data should be collected [18].

There are two main methods of sampling web pages, vertical sampling and horizontal sampling. Vertical sampling is based around sampling web pages based on domain names. An example of vertical sampling would be to only sample web pages from selected countries based on their top-level domain such as .se or .de. Second level vertical sampling is a sub- division to vertical sampling which instead of using the top-level domain to select which domains to sample it uses the name of an organization to select domains [18].

Horizontal sampling is used when the crawler should not be restricted by domain names. For example when the crawler is set to visit a set of domains based on the visitation-patterns from a set of users [18].

### Properties of a crawler

A crawler is constructed around four properties *selection policy*, *re-visit policy*, *politeness policy* and *parallelization policy*. These four properties determine the actions of the crawler and should be defined to fulfill its original purposes [18].

The first property is the selection policy which determines which hyperlinks should be added to the crawl frontier [18].

The second property is the re-visit policy which states when to check for changes to the pages. For a crawler that is continually parsing a very large number of domains and pages in the purpose of discovering new content, the re-visit policy has to be very carefully set to minimize the time that new information is available but not yet parsed by the crawler [18].

The third property is the politeness policy. It states how the crawler should avoid overloading the domains it parses [18]. The politeness policy has to be specified to the capabilities of the domains the crawler will parse and to the amount of domains the crawler will parse. If the crawl frontier is filled with several different domains the risks that the crawler will overload any servers which the domains are stored on is relatively small. The chances of overloading a domain also decreases as size of the domain increases, since a large domain should be able to handle more requests per second [18].

The fourth property is the parallelization policy that states how to coordinate distributed web crawlers. When constructing a continuous web crawler parsing a large amount of information, a parallelization policy is useful to streamline the parsing using methods such as threading. A parallelization policy could also be used to streamline the crawler [18].

To extract the network graph from a domain the information that needs to be checked for is which page links to which page.

# Methods

The network properties chosen to examine for this thesis were the degree distribution, the average path and robustness of the graph. The computational part of the examination as well as the modelling of the properties was done using Python [23] version 2.7.11. A directed, unweighted graph was used to represent the network.

## Extraction of web pages

To do extensive testing, the decision was made to extract the pages and their respective outgoing URLs on kth.se locally. This allowed to avoid unnecessary traffic to kth.se and consider the third crawler property stated in section 2.3.1.

Given the complexity of kth.se and the limited time frame of the thesis, creation of our own crawler from scratch was dismissed. Instead it was opted to use an already existing library, named Scrapy [24], for the Python programming language to extract the web pages. Python was also chosen as the main programming language for this thesis due to its simplicity in handling files, which would be necessary to document many tests.

The files were stored as a manipulated representation of the webpage URL as filename and its outgoing URLs inside the file. To minimize the space consumption of the graph, each URL was assigned an integer to represent it. The URL and its integer representation was stored as a dictionary and the outgoing URLs were stored in a binary file.

## Investigated graph properties

### Degree distribution

The information needed to find the degree distribution of the KTH web was the incoming degree and the outgoing degree of each node in the extracted structure. The incoming degree distribution and the outgoing degree distribution of the network had to be separated as the network of a web page has to be viewed as an unweighted and directed graph.

As the structure was extracted and stored locally in files the structure was easily accessible and the algorithm for finding the outgoing and incoming degree of every node in the network could be constructed.

The program for extracting the degree of every node was constructed using Python. For the outgoing degree the program used the fact that since every outgoing link from a page was stored in each file with the page URL as the filename, each file could be regarded as a node in the network and the number of lines written in the file would be the node’s outgoing degree. For the incoming degree another program was constructed in Python which read each file and kept track of how many URLs it had encountered on each page.

The model used to find the degree distribution for the nodes in the network was 𝑝𝑘

= 𝑁𝑘 .

𝑁

Where 𝑝𝑘 is the distribution of nodes with the degree k, 𝑁𝑘 is the number of nodes with degree k and N is the total number of nodes.

The reason for finding the degree distribution is that it shows a lot about the structure of the network and it is needed to see if the KTH follow the distribution of a random network or if the network is scale-free. When the network is random its degree distribution follows a Poisson- distribution while one property of a scale free network is that the degree distribution follows a power-law distribution.

To view if the properties of the degree distribution of the network matched those of a scale free or a random network the degree distribution of both the outgoing and the incoming degree were used to construct graphs. The incoming degree distribution was also plotted in a graph together with a approximation of the distribution using the power law distribution. The outgoing degree of the network was plotted together with the poisson approximation of the distribution. The poisson distribution was calculated using the Scipy library in Python. The power law approximation was calculated using the formula 𝑝𝑘 = 𝑘−𝑦.

### Average path length

The shortest path in a weighted graph is defined as the smallest sum of the edges’ weights in a path [12]. Between two vertices v1 and v2 the shortest path is the smallest sum of the edges’ weights in a path that connects the two edges. This is usually referred to as the distance between v1 and v2. If no such path connects the two nodes the distance is usually set to an infinite value to denote its nonexistence.

With this definition of the shortest path the average path length in a graph would be defined as the average of all shortest existing paths between every two vertices v1 and v2 in the graph. This problem is generalized often as the “all-pairs shortest paths problem” [12]. As a consequence of this there are no algorithms that calculate the average path length in linear time to the amount of vertices. Therefore a specific amount of random pairs will be selected to approximate the average path length.

An adjacency matrix representation of the graph can be used to calculate the all-pairs shortest path problem [12], but due to the high memory scaling and the large expected network, adjacency lists will be used instead.

#### Breadth first search

Breadth first search (BFS) is an algorithm that traverses all reachable vertices in an unweighted graph, starting from a specific vertex [12]. BFS starts by marking the starting vertex as visited and proceeds to adding all the adjacent vertices to the end of a queue. It then removes the first vertex in the queue and add all its unvisited vertices to the end of the queue and marks the current vertex as visited. The removal and adding of unvisited adjacent vertices is repeated until there is nothing left in the queue.

Due to all the unvisited vertices being added to the end of the queue and removed from the front, the algorithm will search the vertices to which the starting vertex has the shortest path first. This can be utilized to find the length of shortest path between a start and ending vertex by storing the distance to every vertex and stopping the algorithm before the queue is depleted. If the queue is depleted and the end vertex is not found, then there exists no path between them.

## Definition of robustness

In this report the robustness in a network will be defined in terms of the change in average path in targeted and random attacks on vertices. This is done via selection of 10,000 connected pairs of vertices from the start and studying how the average path length scales. A slow scaling of the average path in contrast to the amount of removed vertices will indicate a high robustness and a high scaling indicates a low robustness against targeted and random attacks.

# Results

The crawler created 671,013 files during the crawl of the KTH web, these files contained a total of 23,515,583 links. This section shows the results of the analysis done on the network using these files to create an image of the network. The results from the incoming network and the outgoing network are separated throughout this section.

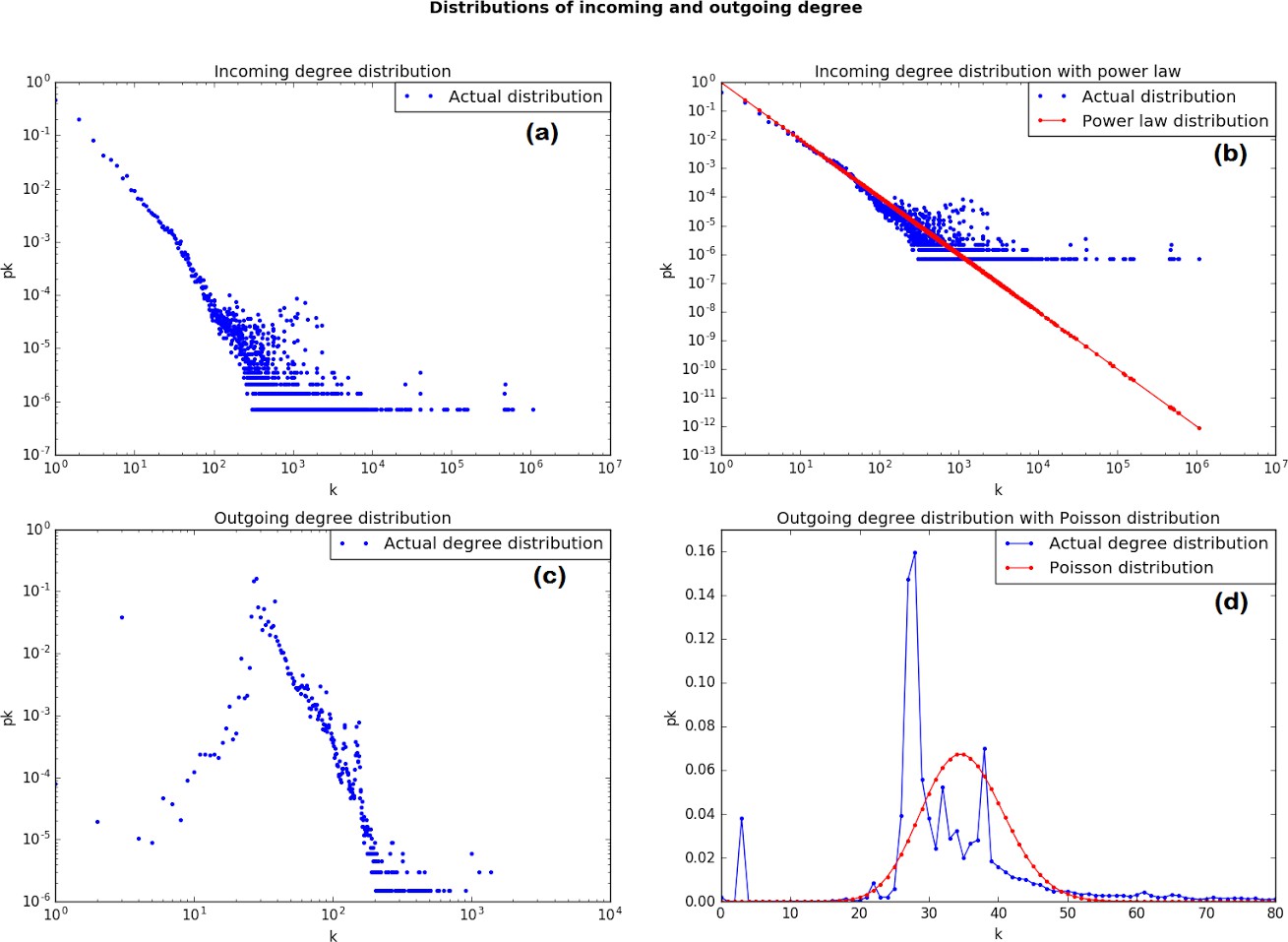
## Degree distribution

The in-degree distribution of the network shows that there are many nodes with few incoming links, and less nodes with many incoming links (Fig. 4.1.1a). This is a common characteristics among scale free networks. The graph also shows an expected cutoff where nodes with a high degree are very few or even unique for their degree. It also shows that the in-degree distribution of the network does not match the degree distribution of a random network.

A power-law distribution was applied to estimate the in-degree distribution (Fig. 4.1.1b). The power law was fitted to the in-degree distribution using the exponent *y = 2*. The graph shows that the power law estimates lower degree nodes very well, but the difference between the estimated and the actual degree grows larger as the degree of the nodes increases.

The out-degree distribution (Fig 4.1.1c) shows some of the characteristics of the in-degree distribution, such as the cutoff at nodes with a higher degree. Note that the outgoing distribution has a similar distribution of low degree nodes to some high degree nodes which is not the case in the in-degree distribution.

An attempt was made to fit a poisson distribution to the out-degree distribution (Fig 4.1.1d). Note that while the poisson distribution peaks at the same time as the actual distribution the approximation is far from the actual distribution at most times. Due to the inconsistent peaks and lows, a power-law distribution would be misrepresentative for the out-degree distribution.



##### Figure 4.1.1:

*The x axis shows k, the degree. The y axis shows pk ,the distribution of the degree k.*

*a) Incoming degree distribution of the KTH web using log-log plot.*

1. *Incoming degree distribution of the KTH web using log-log plot including power law.*
2. *Outgoing degree distribution of the KTH web using log-log plot. The x-axis describes the degree of a webpage and the y-axis describes the amount of web pages with such a degree.*
3. *Outgoing degree distribution of the KTH web including the Poisson distribution. Note, this graph is zoomed in to the area of interest where the majority of nodes are located.*

The main page of the KTH web, kth.se, is the largest hub in the network for incoming links with 1,070,194 incoming links. The second largest hub is kth.se/social with 592,767 incoming links. The tenth largest hub has 484,544 incoming links. The 10 nodes with the highest incoming degree of links are all referred to in the footer of the KTH web.

Outliers also exist not in terms of high incoming degree but in terms of expected occurrence of nodes with a specific incoming degree and the actual occurrence of the incoming degree. One of these outliers are nodes with an incoming degree of 1429 where the actual occurrence is 99. Similar outliers occur at nodes with an incoming degree of 1056, 1115,

1116, 1136, 1511 and 1936. For all these degrees the expected occurrence is much lower than the actual occurrence.

All of the nodes at their respective degree had the same prefix. Nodes with degree 1429 and 1116 consists of pages in the sidebar from the course site for SF1626 and SF1625 respectively. As for degrees 1056, 1115, 1936 all belong to blog pages which refer to each other in the sidebars categories

The node with the highest outgoing degree is the node for kth.se/en/che/site-map which has an outgoing degree of 1378. Outliers in terms of occurrence are nodes with 27 and 28 outgoing links which was also the two most common outgoing degrees. 106,934 nodes had an outgoing degree of 28 while 104,929 nodes had an outgoing degree of 27. 104,929 nodes that has an outgoing degree of 28 and 98,690 nodes with an outgoing degree of 27 all belong to scheduled events in course pages.

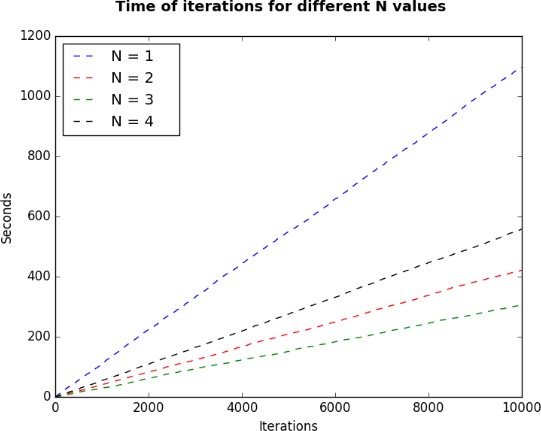
Noteworthy is also the amount of nodes with outgoing degree 3, there are 25,682 nodes with an outgoing degree of 3 which causes a visible peak in the plotted graph 4.1.1(c). Out of these 25,682 nodes 20,110 lead to web pages for rooms on KTH.

## Efficient Average path approximation algorithm by exploiting degree distribution

The mixed distributions of the incoming and outgoing degrees, together with the small world property, resulted in an efficient algorithm for finding the path between two nodes. The algorithm proceeds as follows:

1. Build adjacency lists for both incoming- and outgoing URLs.
2. Create a list *L1* with a length equal to the number of vertices in the graph. The list values should be initialized to a non-positive value marking the node
3. Select two different vertices *start* and *end*
4. Set *L1*[*start*]=0
5. Perform a BFS from *start* on the outgoing URLs until a vertex at a distance *N* is removed from the queue. Mark the these as their distance to *start* \* 1000 in *L1* . If *end* is encountered during the BFS the distance between *start* and *end* will be *L1*[*v1*]+1 , where *v1* is the vertex whose adjacent neighbours are currently checked.
6. Set *L*1[*end*]=1
7. Perform a BFS from *end* on the incoming URLs until a vertex *v1*, adjacent to the vertex *v2* ,is encountered with *L1*[*v1*]>=1000. The distance between *start* and *end* will be *L1*[*v1*] /1000 + *L1*[*v2*]
8. Repeat step 2-7 a certain amount of times

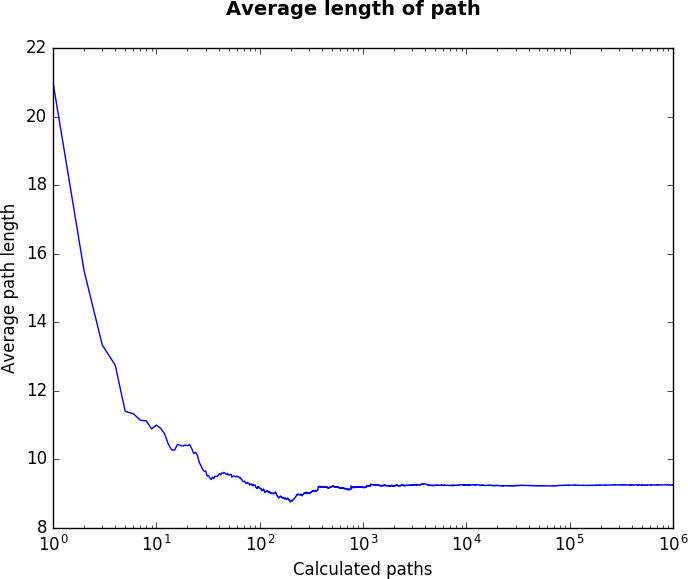
Two time consuming stages of the algorithm were identified, mainly the “reset stage” in step 2 and the “searching stage” from step 5 and 7. The results from the searching stage proved that the value of *N* affected the algorithm’s effectivity, making up for the majority of the search time. This value was decided by running the algorithm with different values on 10,000 pairs of web pages and measuring the time of the searching stage. *N*=3 was proven to be the most consistent and effective value.



***Figure 4.2.1:*** *The search time for different values N for the first BFS. From N=1 it keeps on decreasing until N=3* (green line)*. At N=4 and higher the search time starts increasing.*

## Average path length of the KTH web

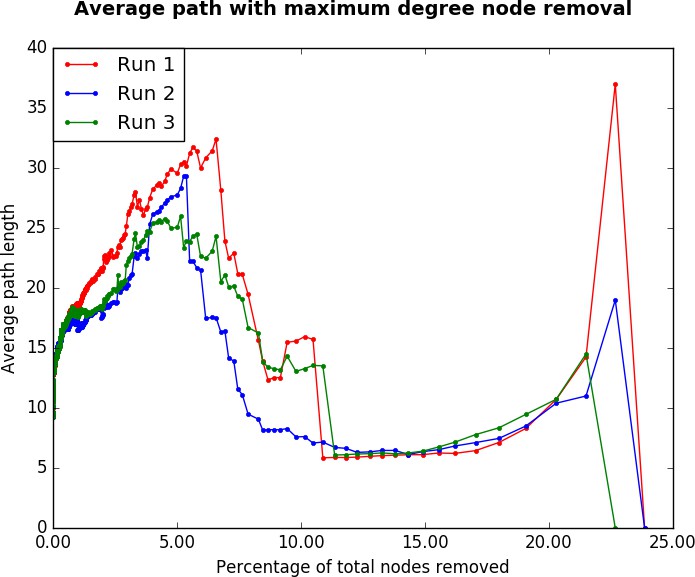
Figure 4.3.1 shows how the length of the average path changes over time when calculated between two randomly selected nodes that exists within the network. The graph shows that the average path converges to roughly 9,25 after many calculated paths. An equally good estimate was achieved after 10,000 iterations.



**Figure 4.3.1:** *Average length of paths between two randomly selected nodes in the network*. *Do note that a good estimate is found in the middle of the figure.*

## Robustness of the network

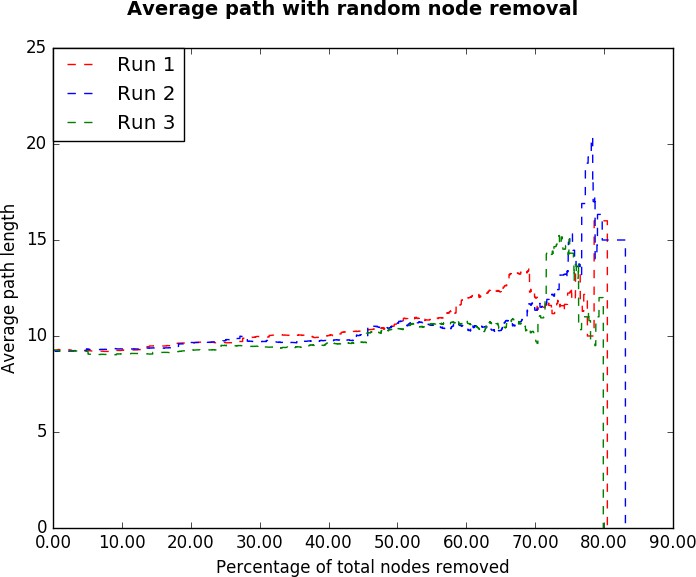
The average path length between 10,000 static randomly selected pairs were studied as nodes were removed from the network (Fig 4.4.1). This test was done in three separate runs represented as different colors. The nodes removed at each step are the ones with the highest degree.



**Figure 4.4.1:** *Average path with targeted node removal.*

As the first 5% of nodes are removed the average path increases and then starts to drop. When about 12% of nodes have been removed there is a large drop in the length of the average path as the nodes of a certain degree are removed. After these nodes have been removed the average path increases in length until there are no existing paths between the pairs.

Figure 4.4.2 shows the effect on average path with random node removal three separate runs represented as different colors. Generally the average path length is slightly increasing up to 70% of removed nodes. After the initial 70% being removed the average path length starts to fluctuate rapidly, leading to longer and adaptable paths persisting. All paths fail after roughly 80-85% of all nodes have been removed.



**Figure 4.4.2:** *Average path with random node removal*

# Discussion

## Results analysis

To answer the initial problem statement, if the degree distributions have a power-law distribution, we analyze figure 4.1.1. As clearly can be seen, the incoming degrees does indeed have a power-law distribution as the results in previous studies. However, the outgoing degrees does not show the characteristics of neither a power-law distribution nor a poisson distribution. What can be concluded from this is that kth.se is a complex random network due to its distribution fitting into several categories. The second part of the initial problem statement about the network’s small-world properties can be answered by examining figure 4.3.1. The average path length of 9,25 in a network of 671,013 pages is much smaller than the average path obtained in the previous study, which was also considered to have small-world properties [4]. A large part of this can probably be attributed to the higher average degree of the web pages.

To analyze the robustness of the network we inspect figure 4.4.1 and 4.4.2. As clearly can be seen the amount of nodes needed to remove to break all paths is much lower for targeted attacks, which is a characteristic observed in scale-free networks. If we were to compare these results to the one obtained in figure 2.2.6, we see that they are very similar when it comes to critical fraction ~0.2 for targeted and ~0.9 for random. Differences include larger fluctuations at the end of each respective figure. One fluctuation that stands out the most is in figure 4.4.1 where it rises after the initial drop at ~10%, whereas in the other report it steadily increases until it suddenly drops. This might indicate that there are few short paths that have good robustness which make up for the otherwise not so stable paths. All in all the kth.se network acts very similar to a scale-free network in terms of robustness.

Modern web design seems to have a large effect on the degree distribution of the network. Because of footers, headers and sidebars, which purpose is to make the site more accessible, there are very few nodes with an outgoing degree of less than 25. There is also the fact that the main page of kth.se is referenced more than once per node, since it is referenced once in the header and once in the breadcrumbs.

The most common outgoing degrees, 27 and 28, are also explained by design choices. As presented in section 4.1 these nodes were mostly made up from event pages. These event pages all had in common that they were scheduled events for courses that used an external site for their course web. The schedule for these courses was still on the KTH web, but the information about the course were on a separate web. This caused the sidebar for the events to be the exact same size for each of these pages and the events therefore had the exact same degree. The difference between the events with an outgoing distribution of 27 and those with 28 were that the ones with 28 also had a reference to the room the event was scheduled in.

Since the outgoing and incoming degree of the network differs a lot on the KTH web, a model for network growth could use separate mechanics for outgoing and incoming edges when adding nodes to an existing network. For incoming edges in the network a model could be based on preferential attachment which generates a long tailed distribution and is commonly used for modeling growth on the [WWW.](http://WWW/)

It was shown that the most common outgoing degrees consisted of nodes that were similar to each other. This suggests that the outgoing network could be modeled using a copy mechanism which duplicates nodes that already exist in the network with mutations. The copy mechanism would have to take into account the footer, header and eventual sidebars when adding nodes to the network to accurately model the network since these have a large effect on the distributions for the network.

## Limitations

This study was restricted to kth.se domain which may not be representative for the WWW as a whole. The crawler used to extract the web pages had irregularities as to which web pages would become files, and which web pages would not be counted as files. Another problem with the crawler was that it selected interpreted URLs differently as to which were stored in the files and which URLs it visited. As a consequence there were some irregularities in the network, leading to roughly 8000 disconnected web pages with 0 incoming URLs. This in turn could cause the results in this study to not reflect the actual network.

Limited amounts of computing resources forced us to make approximations instead of using exact results for the average path. Most of the studied reports [10][7][9] use a different measurement for robustness by studying how the largest component of the graph got reduced in size as vertices were removed. This was not possible due to time constraints for this study.

## Future studies

To determine whether the mixed incoming and outgoing degree distribution was just a trend for kth.se or a part of WWW, further degree distributions from websites needs to be extracted. Accomplishing this requires a crawler that is consistent and follows all the links correctly would be needed. This part should be manually examined at the start to make sure everything was done correctly.

Another interesting aspect to explore would be to create a model for the growth of kth.se which would result in the degree distribution of this study. A model like this might contain some rather selective duplication mechanism, as most menus for each page is very similar, if not completely identical, for many more web pages. Verification of this model would be done via simulations. If the simulated network provides a similar distribution its robustness could be tested more extensively in smaller samples. The robustness analysis could be done in terms of average path length and largest component size.

## Conclusions

This study shows that the WWW still has the small-world property and that the scale-free distribution discovered in 1999 still hold for incoming degrees. However, the results also give an indication that something might have changed for the outgoing degree of web pages on the [WWW.](http://WWW/) Modern web design that includes footers, headers and sidebar is most likely the cause for this change. The modern design is also likely a cause for the low average path of the network. It was also shown that most of the peaks for the outgoing degree could be explained by the existence of many similar nodes. Irregularities in the extracted crawler data may have caused slightly different results. For confirmation of the results the study should be repeated with full access to the KTH web.

# References

1. M Newman(2003). The structure and function of complex networks. *SIAM Reviews* 55: 167-257.
2. A-L Barabasi(2007). The architecture of complexity. *IEEE Control Systems Magazine.*

27(4):33-42

1. A.-L. Barabási. More is Different : Network Science. Lecture presented at; 2013; Nanyang Technological University. URL: <https://www.youtube.com/watch?v=NCq1kpuCUq4&feature=youtu.be&t=4790> . Accessed: 2016-04-10
2. R. Albert, H. Jeong and A.-L. Barabási, “Diameter of the World Wide Web,” Nature, vol. 401, pp. 130–131, Sept. 1999.
3. R. Govidan, H. Tangmunarunkit (2000). Heuristics for Internet Map Discovery. *IEEE INFOCOM 2000*, vol. 3. 1371-1380. ISSN: 0743-166X . DOI : 10.1109/INFCOM.2000.832534. URL:

<http://ieeexplore.ieee.org.focus.lib.kth.se/xpl/articleDetails.jsp?arnumber=832534> .

1. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.-L. Barabási (2000). The large-scale organization of metabolic networks. *Nature*. Vol. 407. 651-654. ISSN: 0028-0836. DOI:10.1038/35036627 . URL:

<http://www.nature.com.focus.lib.kth.se/nature/journal/v407/n6804/full/407651a0.html> .

1. A.-L. Barabási and R. Albert(2002). Statistical mechanics of complex networks. *Rev. Mod. Phys*. vol. 74. 47–97.
2. I. Ispolatov, P. L. Krapivsky, and A. Yuryev (2005). Duplication-divergence model of protein interaction network. *Physical Review E* . Vol 71(6). 06911-1 - 06911-8 . DOI: <http://dx.doi.org.focus.lib.kth.se/10.1103/PhysRevE.71.061911> . URL: <http://journals.aps.org.focus.lib.kth.se/pre/abstract/10.1103/PhysRevE.71.061911>
3. R. Albert, H. Jeong, and A.-L. Barabási (2000). The Internet’s Achilles’ heel: Error and attack tolerance in complex networks. *Nature*, vol. 406. 378–382.
4. W. Li, Y. Jia-Ren, L. Zi-Ran, and Z. Jian-Guo (2007). Attack Vulnerability of Network with Duplication-Divergence Mechanism. *Communications in Theoretical Physics*. Vol 48(4). 754- 758. DOI: <http://dx.doi.org/10.1088/0253-6102/48/4/038> .
5. A.-L. Barabási (2013). Network Science. *The Royal Society*. Vol 371. ISSN:1471-2962 . DOI: <http://dx.doi.org/10.1098/rsta.2012.0375> . URL: <http://rsta.royalsocietypublishing.org.focus.lib.kth.se/content/371/1987/20120375>
6. Michael T. Goodrich, Roberto Tamassia. *Algorithm Design: Foundations, Analysis, and Internet Examples*. John Wiley & Sons, Inc. USA, 2012. ISBN: 978-0-471-38365-9
7. Norman L. Biggs. *Discrete mathematics: Second edition.* Oxford University Press Inc,

Great Britain, 2003. ISBN: 9780198507178

1. Jean Gallier (2015). Spectral Theory of Unsigned and Signed Graphs Applications to Graph Clustering: a Survey. *University of Pennsylvania*. Available from: <http://www.cis.upenn.edu/~jean/spectral-graph-notes.pdf>
2. S. Milgram (1967). The small world problem. *Psychol. Today*. vol. 1. 60–67.
3. D.Watts ,S. Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature*. Vol

393. 440-442. ISSN: 0028-0836 . DOI: 10.1038/30918 . URL:

<http://www.nature.com.focus.lib.kth.se/nature/journal/v393/n6684/full/393440a0.html>

1. A.-L. Barabási, A. László , R. Albert, J. Hawoong (1999). Statistical Mechanics and its Applications, *Physica A* . Vol.272(1). 173-187.
2. Castillo C (2004). Effective Web Crawling. *ACM SIGIR Forum 55 Vol.39. 55-56*. <http://chato.cl/research/crawling_thesis>
3. Google (2011). Facts about Google and Competition. http://web.archive.org/web/20111104131332/<http://www.google.com/competition/howgoogles> earchworks.html
4. Arasu, A, Novak, J, Tomkins, A. and Tomlin, J. (2002). PageRank computation and the structure of the web: Experiments and algorithms. pp. 107–117.
5. Perra N and Fortunato S. (2008). Spectral centrality measures in complex networks.

Phys. Rev. E 78 (3): 36107

1. Outbrain (2011). Content discovery and engagement report. URL: <http://www.outbrain.com/blog/2011/04/outbrain-content-discovery-report.html> . Accessed: 2016-03-15
2. Python Software Foundation (2016). URL: https://[www.python.org/](http://www.python.org/) . Accessed: 2016- 04-17
3. Scrapinghub (2016). URL: <http://scrapy.org/> . Accessed: 2016-04-17



3