6. Random Graphs, Small-Worlds and Scale-Free Networks

Katharina Anna Lehmann, Michael Kaufmann
(University of Tübingen)

6.1 Introduction

In this chapter we will introduce two famous network models that arose much interest in recent years: The small-world model of Duncan Watts and Steven Strogatz [615] and scale-free or power-law networks, first presented by the Faloutsos brethren [201] and filled with life by a model of Albert-László Barabási and Réka Alberts [60]. These models describe some structural aspects of most real-world networks. The most prevalent network structure of small-world networks is a local mesh-like part combined with some random edges that make the network small.

The preceding chapters sketched the field of Peer-to-Peer concepts and applications. The field and its history are deeply intertwined with the area of distributed computing and sometimes overlaps with concepts from client-server systems and ad hoc networks. To set a clear foundation we base this chapter on the following, quite general definition of Peer-to-Peer systems:

Definition 6.1.1. *Peer-to-Peer Systems*
A Peer-to-Peer system consists of computing elements that are:

(1) connected by a network,
(2) addressable in a unique way, and
(3) share a common communication protocol.

All computing elements, synonymously called nodes or peers, have comparable roles and share responsibility and costs for resources.

The functions of Peer-to-Peer systems are manifold. They may be coarsely subsumed under **communication of information, sharing services, and sharing resources**. To implement these functions, the system has to provide some infrastructure. What are the requirements to make a Peer-to-Peer infrastructure useful? Here, we will concentrate on the following four conditions:

Condition 1 **Decentrality**
One inherent claim of the Peer-to-Peer idea is that there is no central point in which all information about the system, data and users is stored. If there is no central organizing element

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 57-76, 2005.

in a system, each node needs some built-in rules with which it can, e.g., join the system, route information to others, or make queries.

Condition 2 **Structure**
To share data and other resources efficiently it is helpful if the system is structured in a way that improves searching and facilitates routing.

Condition 3 **Reliability despite Dynamic Changes**
Nearly all Peer-to-Peer systems are under constant changes: nodes join the system, others leave it, and some are just temporarily unavailable. This necessitates elaborate mechanisms to stabilize important system properties like the diameter and connectivity of the system.

Condition 4 **Scalability**
One effect of decentralization is that the number of nodes may be arbitrarily large. Nonetheless, the system should be able to service the needs of all nodes in an efficient and fast way.

These four conditions are especially interesting and resemble features of social networks: A social network is inherently self-organized but nonetheless structured. Despite the fact that people are born and die, many global structural properties of social networks are stable. Further, there seems to be no limit to the number of humans that can take part in our global social network - only with respect to the network's structure, of course.
Other 'real' networks that have evolved over time show similar properties, like the Internet, metabolic networks, or the WWW [201, 60, 324]. What makes evolving, decentralized networks structured and stable despite their dynamics? Recent research has revealed that there are two important properties of evolving networks that help to satisfy the above conditions: The first property is termed the *small-world effect*, the second property is a *scale-free degree distribution*. Networks with a small-world effect are called **small-worlds** and networks with a scale-free degree distribution are called **Scale-Free Networks**. Peer-to-peer systems that create overlay networks with these features are likely to satisfy the conditions without further concern. This chapter presents both properties, presents families of networks that display these features and the corresponding decentralized models to create them. To do so, we first define some notions in graph theory in Sec. 6.2. In Sec. 6.3 we will briefly describe how the analysis of social networks was challenged by a series of elegant experiments in the 1960's. Sec. 6.4 presents the most common models that answer the questions raised by those experiments. Specifically, these are models that create families of Small Worlds and Scale-Free networks. In Sec. 6.5 we describe approaches that build small-worlds or Scale-Free overlay networks in Peer-to-Peer systems. In this section we will present the most important properties of each of the network families which influence the per-

formance of Peer-to-Peer systems. In Sec. 6.6 we will summarize the results and discuss possible improvements.

6.2 Definitions

Let $V = \{1, 2, 3, \ldots, n\}$ be the set of all n *nodes*, or *peers*, in the system. Each node is identified and addressable by its number. The underlying network makes it possible to route a message from each node to any other node. Because of the decentralized nature of a Peer-to-Peer system, not every node v is required to store routing information to each and every other node. The set of nodes over which node v will route outgoing messages is denoted by $N(v)$ and called the *neighbors of* v. Every Peer-to-Peer network can be associated with a graph $G = (V, E)$. E is the set of edges $e = (i, j)$ where j is a neighbor of i, i.e., there is at least one entry in the routing table of i that uses j as the next node. For edge $e = (i, j)$, i is the *source node* and j is the *target node*. The number of edges is denoted by m. G is sometimes called the *overlay network* of a Peer-to-Peer system. The edges might be weighted, e.g., with the number of entries that use j as the next node or the cost for the traverse of this edge. All edges are *directed*.

The set of edges can also be represented in the *adjacency matrix* $A(G)$ with dimension $n \times n$. a_{ij} is 1 if and only if the edge $e = (i, j) \in E$. If edges are weighted with a weight function $\omega : E \to \mathbb{R}$ then a_{ij} is commonly given by $\omega(e = (i, j))$ if $e = (i, j)$ and zero otherwise. The set of *eigenvectors* and *eigenvalues* of a matrix is defined as the set of all vectors x and real numbers λ such that:

$$Ax = \lambda x \tag{6.1}$$

The *outdegree* $k_o(v)$ of a node v is defined as the number of neighbors it has: $k_o(v) = |N(v)|$. The *indegree* $k_i(v)$ is defined as the number of neighbor sets in which v is an element: $k_i(v) = \sum_{w \in V}[v \in N(w)]$. The Boolean expression in brackets is given in Iverson-notation (see [257]) and evaluates to 1 if the expression is true and to zero otherwise. The *degree* $k(v)$ of a node v is defined as the sum of indegree and outdegree. A *path* $P(i, j)$ from node i to node j is defined as a subset $P \subseteq E$ of edges $\{e_1, e_2, \ldots, e_k\}$ where $e_1 = (i, v_1)$, $e_k = (v_{k-1}, j)$ and $\forall\ 1 < l < k$: $e_l = (v_{l-1}, l)$. The *path length* of a path P is defined as the number of edges in it. If the edge set is weighted with a weight function $\omega : E \to \mathbb{R}$, then the path length $L(P(i, j)$ of a path $P(i, j)$ of two nodes i and j is defined as:

$$L(P(i, j)) = \sum_{e \in P_s(i,j)} \omega(e) \tag{6.2}$$

In the following, we will use the first definition to reduce complexity. All further definitions can be easily transformed for weighted graphs.

Any path with minimal length between two nodes is a *shortest path* between these nodes. The *distance* $d(i,j)$ between two nodes is the length of any shortest path between them. The *diameter* $D(G)$ of a graph G is defined as the maximal distance between any two nodes in the graph:

$$D(G) = \max_{(i,j)\in V\times V} d(i,j) \tag{6.3}$$

The *average path length* $D_{\oslash}(G)$ of a graph G is defined as the sum of the distances over all pairs of nodes divided by the number of all pairs of nodes:

$$D_{\oslash}(G) = \frac{\sum_{(i,j)\in V\times V} d(i,j)}{n\cdot(n-1)} \tag{6.4}$$

A graph is *connected* if there is at least one path between every two nodes. A graph is k-connected if the removal of any set with $k-1$ nodes leaves the graph connected. Let $V_C \subseteq V$ be a subset of nodes. The *induced graph* $G(V_C) = (V_C, E_C)$ is defined as the graph with the following edge set E_C: $E_C = \{e = (i,j) | i,j \in V_C\}$. An induced graph is a *(simple) component* if it is connected.

The set of edges in a graph is formally a relation $\mathcal{R} \subseteq V \times V$ on the set of possible edges. A *network family* is an (infinite) set of graphs with the same relation. Normally, this relation is given as an algorithm that decides which edges are added to the graph and which are not.

6.3 The Riddle – Analysis of Real Networks

In the 1960's Stanley Milgram conducted a series of interesting experiments [413, 596] that posed new questions about how humans are organized into social networks: He prepared letters to a friend of his and sent them to people chosen randomly in Kansas and Nebraska (cf. Fig. 6.1). They were asked to deliver the letter but they got no more information than the name of the recipient, his profession (stock broker) and the town he lived in (Boston). Furthermore, they were asked to deliver the letter in a special way: Instead of using the address, the letter should be given to someone they knew on a first-name basis and which they thought to be 'closer' to the recipient in any way. Eventually, some of the letters reached the broker and, surprisingly, these letters did not need many steps to find him. On average, it were not more than six steps [1]. This result was very surprising because social networks are dominated by relationships to people that live and work near to us. This was especially true in a time when there was no Internet, and where cars and airplanes were too expensive to be an everyday means of transportation for

[1] Most interestingly, the concept of **six degrees of separation** was already mentioned in a short story entitled 'Chains' by the Hungarian writer Karinthy in 1929 [339]

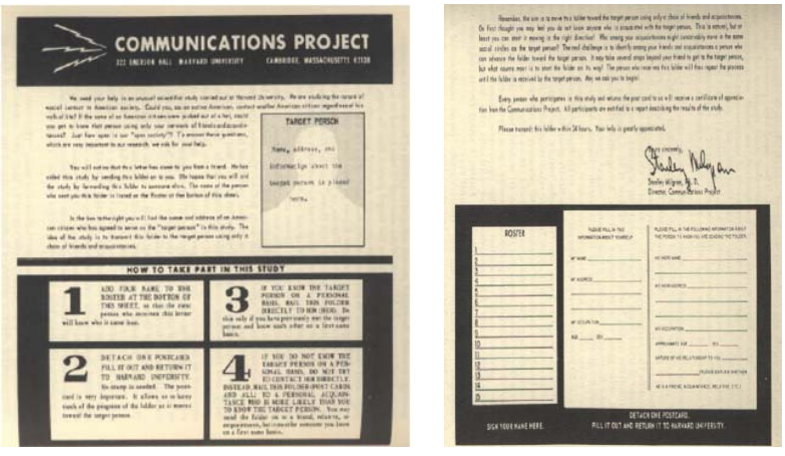

Fig. 6.1: Letter sent in the Milgram Experiment

most people. In terms of graph theory the result signifies that the diameter of social networks is quite small despite their dense local structure. What kind of network model can reproduce this special combination of properties? This is the riddle that was not to be solved until the 1990s. In the following sections we will describe the most important approaches with which social and other evolving networks are modeled today. We will show that some of the features of these networks are interesting for Peer-to-Peer applications and present ideas about how their favored properties can be transferred to Peer-to-Peer overlay networks.

6.4 Families and Models for Random Graphs, Small-Worlds and Scale-Free Networks

Historically, random graphs form the first family of networks that were intensely studied. Since many Peer-to-Peer applications choose neighbors more or less randomly, like Gnutella, this model is also valuable for the analysis of Peer-to-Peer systems.

6.4.1 Random Graphs

The analysis of social relationships as graphs can be traced back to the 1950s [614]. At the same time, the first graph models, concerning random graphs, were introduced. They were so successful that they were used as simulation models for very different networks over the following 30 years. Random graphs

were introduced independently by Gilbert [245] and Erdős and Renyi [464]. We will first present the model of Erdős and Renyi, following [84].

Erdős and Renyi's Notation

$\mathcal{G}_{n,m}$ denotes the space of all $\binom{N}{m}$ graphs with n nodes and m edges where $N = \binom{n}{2}$ is the number of possible edges between n nodes. This set can be transformed into a probability space by taking the elements of $\mathcal{G}_{n,m}$ with the same probability. An instance of $\mathcal{G}_{n,m}$, drawn uniformly at random, is denoted by $G_{n,m}$. Since the whole model is based on stochastic processes, we can only give probabilistic statements about **expected properties**, i.e., we say that $G_{n,m}$ shows property P with a high probability if:

$$Pr(G_{n,m} \text{ has } P) \to 1 \text{ for } n. \to \infty \tag{6.5}$$

It is important to note here that this is only interesting if m is a function of n. If it is constant for all n, the graph will be disconnected for large n and most interesting properties like the average degree or the connectivity of the graph will vanish with $n \to \infty$. This leads to a question regarding the first important property of random graphs: When will a random graph be connected with high probability? The following theorem gives the important relationship between m and the connectedness of the resulting graph ([464]):

Theorem 6.4.1. *Connectedness of random graphs*
Let $m_\omega = \frac{n}{2}(\log n + \gamma)$ where $\gamma = \gamma(n)$ is a function of n. If $\gamma \to -\infty$, then a typical G_{n,m_γ} is disconnected, whereas if $\gamma \to \infty$, a typical G_{n,m_γ} is connected.

This theorem can often be found in the following form: If we ensure that the average degree of nodes is $\Omega(\log n)$, then the random graph will be connected with high probability. In the analysis of most network models it is shown that the average degree grows with $O(\log n)$ which also fulfills the above given theorem, if we set γ to $\log n$.

Gilberts Model

A totally different approach was given by Gilbert in [245]. A graph $G_{n,p}$ is defined as a graph in which the probability that an edge $e = (v, w)$ exists is p. This definition simultaneously gives a construction algorithm: For each possible edge, a random number between 0 and 1 is drawn. Whenever this number is smaller than p, the edge is added to the graph. Again, p can be a function of n, though in this case, families with constant p are also interesting to study.

Connection Between Both Random Graph Models

For $M \sim pN$ the two models $\mathcal{G}_{n,M}$ and $G_{n,p}$ are almost interchangeable [84].

Basic Results for Classical Random Graphs

Here, we will just review some of the important results for Random Graphs that are interesting in comparison with small-worlds and Scale-Free Networks, cited from [80]. We use the Gilbert notation.

For the first theorem cited, the random graph is built up sequentially, by adding random edges one at a time. Analyzing the connectivity of the evolving graph, we can make an interesting observation: After having added approximately $n/2$ edges, we get a giant connected component with a size of $\Theta(n)$ as stated in the following theorem.

Theorem 6.4.2. *Giant Connected Component*
Let $c > 0$ be a constant and $p = c/n$. If $c < 1$ every component of $G_{n,p}$ has order $O(\log n)$ with high probability. If $c > 1$ then there will be one component with high probability that has a size of $(f(c) + O(1)) \cdot n$, where $f(c) > 0$. All other components have size $O(\log n)$ [84].

This theorem is easy to remember and nonetheless surprising: The giant connected component emerges with high probability when the average degree is about **one**.

The next property concerns the degree distribution: If one node is drawn randomly from V, how high is the probability $P(k)$ that it has degree k? In random graphs the degree distribution is described as a Poisson-distribution $P(k) = \frac{c^k e^{-c}}{k!}$ as stated in the following theorem:

Theorem 6.4.3. *Degree distribution*
Let X_k be the number of nodes with degree k in $G_{n,p}$. Let c be a constant with $c > 0$ and $p = c/n$. Then, for $k = 0, 1, 2 \ldots$

$$Pr\left((1-\epsilon)\frac{c^k e^{-c}}{k!} \leq \frac{X_k}{n} \leq (1+\epsilon)\frac{c^k e^{-c}}{k!}\right) \tag{6.6}$$

as $n \to \infty$ [84].

This can easily be seen by the following argument: First, we can construct $G_{n,p}$ in a Bernoulli experiment with $\binom{n}{2}$ variables $X_{ij}, i \neq j, i, j \in V$ that are 1 with probability p. The degree of node i is the sum of all variables X_{ij} and for reasonably small p and $n \to \infty$, the degree can be described by a Poisson distribution.

The next question to be answered is the diameter of $G_{n,p}$. It is given by the following theorem:

Theorem 6.4.4. *Diameter of $G_{n,p}$*
If $pn/\log n \to \infty$ and $\log n/\log(pn) \to \infty$ then the diameter of $G_{n,p}$ is asymptotic to $\log n/\log(pn)$ with high probability.

The last property presented here is the expected *clustering coefficient* for random graphs. The clustering coefficient measures how many edges are between neighbors of node i divided by the maximum possible number of edges between them. Thus, it measures how 'clique-like' the neighborhood of node i is where a clique denotes a subgraph in which all nodes are connected to all other nodes. Let $E(N(i))$ denote the number of edges between neighbors of node i. Then, the clustering coefficient $C(i)$ is defined as:

$$C(i) = \frac{E(N(i))}{d(i)(d(i)-1)} \tag{6.7}$$

The clustering coefficient can also be interpreted as the probability that two randomly drawn neighbors of i are themselves neighbors. Seen under this perspective the following theorem is easily proven:

Theorem 6.4.5. *Clustering Coefficient of random graphs*
The clustering coefficient of a random graph is asymptotically equal to p with high probability.

Random graphs were very famous for a long time for two reasons: Many of their properties are exactly solvable in a rigorous analysis. They can be exactly defined and varied in many different ways. Second, they provide a much richer field of application than the other network model that was popular at the time, i.e., regular graphs in which every node has the same degree, such as a lattice. No one doubted that social networks cannot be exactly random, but as long as some of their properties were well described by it, it seemed that random graphs were an easy and useful way to model all kind of different networks.

6.4.2 Small-Worlds – The Riddle's First Solution

Despite the excitement that followed the Milgram experiment there was no convincing network model generating a network that is locally highly clustered and at the same time has a small diameter until 1998. Then, Watts and Strogatz [615] analyzed three different kinds of real networks: A film collaboration network in which two actors are connected by an undirected edge whenever they have acted together in at least one film, the neural network of the worm *C. elegans*, and the power grid of the United States. For each of these networks they measured the average path length in the graph and compared it with a random graph with the same number of nodes and edges. The average path length was in each case slightly higher but clearly within the

same order of magnitude. On the other hand one could see that the real networks were much more densely connected on a local level than the random ones. To measure this density, the authors introduced a new measure, the *clustering coefficient* which we have already defined in Equation 6.7. Watts and Strogatz compared the average clustering coefficient of these real networks with the corresponding random networks: The clustering coefficients were at least ten times higher in real networks and for the film collaboration network the factor is more than 1000. With this analysis the surprising result of Milgram's work could be made more intelligible: Real networks have nearly the same diameter as Random Graphs and at the same time show a high, local clustering.

	$D_{\oslash}$(real)	$D_{\oslash}$(random)	C(real)	C(random)
Film collaboration	3.65	2.99	0.79	0.00027
Power grid	18.7	12.4	0.08	0.005
C. elegans	2.65	2.25	0.28	0.05

Table 6.1: Average path length $D_{\oslash}$ and average clustering coefficient C for three real networks, compared with random graphs that have the same number of nodes and the same average degree. The first network represents actors that are connected by an edge if they have contributed to at least one film together, the second is defined as the set of all generators, transformers and substations that are connected by high voltage transmission lines. The neural network of *C. elegans* displays all neurons and considers them as connected if they share a synapse or gap junction. All three networks show the small-world phenomenon, with an average path length comparable to that of the corresponding random graph and a clustering coefficient that is considerably larger than in the random graphs ([615]).

With this, small-worlds are defined as networks with a dense, local structure, evaluated by the clustering coefficient, and a small diameter that is comparable to that of a random graph with the same number of nodes and edges. Watts and Strogatz introduced a very simple network model that is able to reproduce this behavior. It starts with a chordal ring: Nodes are numbered from 1 to n and placed on a circle. Then, every node is connected with its k clockwise next neighbors (Fig. 6.2)

This ring is traversed and for every encountered edge a random number between zero and one is drawn. If it is smaller than a given constant $0 \leq p \leq 1$ the edge will be rewired. The rewiring is done by drawing uniformly at random a new target node from the set of all nodes V, deleting the old edge and inserting the new edge between the old source and the new target node. It is important to preclude duplicate edges in this process. If p is small,

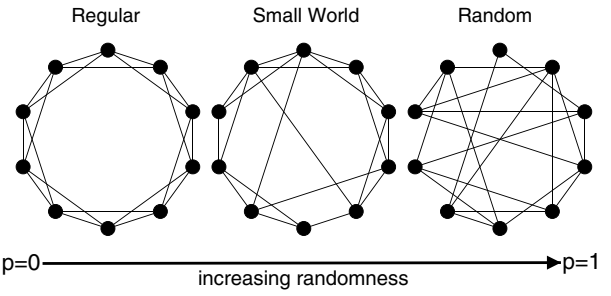

Fig. 6.2: The small-world model introduced by Watts and Strogatz [615] starts with a chordal ring in which n nodes are placed on a circle and connected with their k clockwise next neighbors (here, $k = 2$). With probability p every edge can be rewired once. The rewiring is done by choosing uniformly at random a new target node, such that the old edge is removed and the new one connects the old source node with the new target node. The figure shows that as p grows the model can be tuned between total regularity and total randomness. With sufficiently small p it is possible to maintain the local structure and yet provide an overall small diameter. This state thus displays the properties of small-worlds as they can be found in reality.

almost no edges will be rewired and the local structure is nearly completely preserved. If p is near to 1 the graph produced is claimed to be a random graph with a small average path length. Interesting are the states in between these two extremes. Fig. 6.3 shows the dependency of the clustering coefficient and average path length on p for a graph with 5000 nodes. Clearly, even a quite small p of about 0.005 is sufficient to reduce the diameter so much that it resembles the diameter in the corresponding random graph without losing the local structure that is measured with the clustering coefficient.

Viewed from another perspective, the findings of Watts and Strogatz indicate that a small number of random edges decreases the average path length significantly since they can be viewed as 'short-cuts' spanning the regular graph. With this model a part of the riddle regarding real networks was solved.

In Sect. 6.5 we will present some more properties of small-worlds that are especially interesting for Peer-to-Peer applications. In Subsect. 6.5.1 we will present a more generalized model of Small World Networks in multi-dimensional spaces, introduced by Kleinberg in [353, 354]. But despite the immediate success of the small-world model of Watts and Strogatz the riddle was only partly solved, as would soon become clear.

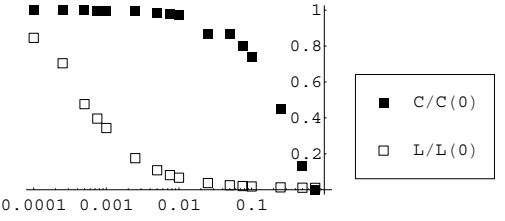

Fig. 6.3: The diagram shows the dependency of the clustering coefficient C and the average path length L on the rewiring probability p. For each probability ten different small-worlds with 5000 nodes have been simulated. The clustering coefficient of the small-world after the rewiring phase was divided by the clustering coefficient of the chordal ring before rewiring. Analogously, the average path length is given in relation to the average path length before the rewiring. It can be clearly seen that a small rewiring probability of approximately 0.005 is sufficient to reduce the average path length to 1/10 without decreasing the clustering coefficient by more than 1.5%.

6.4.3 Scale-Free Networks: How the Rich Get Richer

Although the small-world model explains how two seemingly contradictory properties can be merged into one model, it is clear that it cannot explain *how* these properties emerge in real, evolving networks. Practically no real network can be represented by a ring topology where some edges exist between two randomly chosen nodes. This is a drawback of the model.

But there is another, more significant property missing in the model: In the small-world model nearly every node has the same degree and it is very improbable that a node with a very high degree will emerge. In real random graphs the probability of drawing a node with degree k is proportional to $c^k/k!$. How probable is it to find nodes with a very high degree in real networks?

In 1999, three brothers, Michael, Petros and Christos Faloutsos, made a very extensive analysis of the Internet backbone [201]. They were interested in the following questions: "What does the Internet look like? Are there any topological properties that don't change in time? How will it look like a year from now? How can I generate Internet-like graphs for my simulations?". They examined the inter-domain topology of the Internet from the end of 1997 to the end of 1998. In this phase the Internet grew about 45% in size (number of routers). They found four properties of these networks that follow a power law:

E 1) First, a list of all existing outdegrees was made and sorted. The 'rank' r_i of a node i is defined as its place in the list according to its outdegree $k_o(i)$. The outdegree $k_o(i)$ of a node i is proportional to its rank, to the power of a constant $\mathcal{R}$:

$$k_o(i) \propto r_i^{\mathcal{R}} \tag{6.8}$$

E 2) The number of nodes f_k with the same outdegree k_o is proportional to the outdegree to the power of a constant $\mathcal{O}$:

$$f_{k_o} \propto k_o^{\mathcal{O}} \tag{6.9}$$

E 3) The eigenvalues, λ_j, of a graph are proportional to the order, j, to the power of a constant, $\mathcal{E}$:

$$\lambda_j \propto j^{\mathcal{E}} \tag{6.10}$$

E 4) The total number $P(d)$ of pairs of nodes (i, j) within a distance $d(i, j) \leq d$ is proportional to the degree d to the power of a constant $\mathcal{H}$:

$$P(d) \propto d^{\mathcal{H}} \tag{6.11}$$

This last property is more approximative than the other properties but is nonetheless useful as the authors show in their paper [201].

After the authors had found this self-organizing structure, they asked in their discussion "Why would such an unruly entity like the Internet follow any statistical regularities?". The answer to this question was given by an elegant model of Barabási and Albert in the same year[60]. They examined a part of the World Wide Web (WWW) [60] (see also [20]) and displayed the result as a graph. In this graph, all visited pages were represented as nodes, and two pages were connected by a directed edge (i, j) if page i had a link pointing to page j. In this graph the number of nodes with a given degree was calculated. By dividing it by the number of nodes in the graph, the probability $P(k)$ of drawing uniformly at random a node with degree k is computed. The authors observed that the probability $P(k)$ is proportional to k to the power of a constant γ (similar to E3 above):

$$P(k) \propto k^{-\gamma} \tag{6.12}$$

Networks with this property are called *Scale-Free Networks*, or sometimes *Power-Law Networks*. Barabási and Albert also examined the film collaboration network and power-grid of the USA and found the same property there. To model this property they introduced the following new model that differs in two important aspects from the small-world model and random graph model:

The Barabási-Albert-Model

1. The network grows in time.
2. A new node joining the network will have preferences to whom it wants to be connected. This *preferential attachment* is modeled in the following way: Each new node i wants to connect to m_0 other nodes that are

already in the network. The probability $\Pi_t(j)$ that some old node j gets one of the m edges is proportional to its current degree $k_t(j)$ at time t:

$$\Pi(j) = \frac{k_t(j)}{\sum_{v \in V} k_t(v)} \tag{6.13}$$

$$= \frac{k_t(j)}{2 \cdot m_t} \tag{6.14}$$

with m_t being the number of edges in the graph at time t.

Thus, the network model works as follows:

1. Begin with a small network of at least m_0 nodes and some edges.
2. In each simulation step add one node. For each of its m_0 edges draw one of the nodes j that are already in the graph, each with probability $\Pi(j)$.

It should be clear that this algorithm is not a model in the mathematical sense [80] but rather defines a family of possible implementations. Later, Albert and Barabási could show in [22, 19] that the only requirement for the emergence of a scale-free behavior is that the probability of gaining a new edge is proportional to the degree of a node in each timestep. Thus, it is sufficient that any network model show this preferential attachment in order to generate scale-free networks. This property can be easily remembered as a behavior in which 'the rich get richer'.
To date, many different variants of network models that generate scale-free networks have been published: A mathematical model more precisely defined than the Barabási-Albert-model is the *linearized chord diagram*, introduced in [79]. Here, two groups provide each node with an initial attractiveness that increases the probability of being chosen by a constant value [180, 175]. A quite complicated but powerful model with many parameters was given in [132].

A model that is simple to adapt to Peer-to-Peer systems was first introduced by Kumar et al. [369] for web graphs, and independently by Vazquez et al. [605] and Pastor-Satorras et al. [477] for modeling protein interaction networks: In each timestep of this model, one of the existing nodes is cloned with all the links to other nodes and the clones are connected to each other. Then, both clones lose some edges at random with a very small probability and gain as many new edges to new, randomly drawn target nodes. It can be easily shown that the probability of node j getting a new node in timestep t is proportional to its degree at that time: The more edges it has, the more probable it is that one of its neighbors is chosen to be cloned. If one of the neighbors is cloned, the edge to j is copied and thus the degree of j is increased by 1. Thus, this model shows preferential attachment and the resulting networks are scale-free with respect to the degree distribution.

In the following we want to discuss some of the properties of Small Worlds and Scale-Free Networks that are interesting for Peer-to-Peer systems.

6.5 Applications to Peer-to-Peer Systems

6.5.1 Navigating in Small-Worlds

Jon Kleinberg was also intrigued by the experiments conducted by Milgram and the simple small-world model given by Watts and Strogatz. This latter model explained why there exist short paths in a social network where only a small fraction of edges are actually random. But Kleinberg saw that there was more to it: he asked himself why people can *find* these short paths: "Why should arbitrary pairs of strangers be able to *find* short chains of acquaintances that link them together?" [354]
This question is totally different from the question of why short paths *exist* in a network. It is easy to invent a network with low average path length but where it is impossibility to find those short paths: The difficulty arises if every node just has local information. This is certainly the case in social networks: here, every person just knows a very small number of people on a first-name basis. Of these, one may also know some more data, like the profession, address, hobbies, and so on. When challenged with the task of sending a letter to a stranger via acquaintances, people choose the one friend that they think to be 'next' to the recipient. Milgram's results showed experimentally that the first steps of the letter were the largest (geographically) while later they became shorter as they were closing in on the target area [413].

The aim of Kleinberg was thus to find a family of simple networks with small average path length in which decentralized algorithms are able to find short paths. We will first begin with the underlying network model. It is applicable to multi-dimensional spaces, but here it will be represented in a twodimensional space for simplicity.

The Kleinberg Small-World Model

The model starts with a set of grid points in an $n \times n$ square. Each node i is identified by the two coordinates x_i, y_i that define its position $P(i)$ in the grid. The distance $d'(i,j)$ is here defined as the number of 'lattice steps' separating them:

$$d'(i,j) = |x_i - x_j| + |y_i - y_j| \tag{6.15}$$

The set of (directed) edges is constructed in two parts:

1. First, every node i is connected with all nodes j that are within distance $d'(i,j) \leq q$ for some given integer q.
2. Second, for each node i q additional edges are built. The probability that the i^{th} directed edge has endpoint j is proportional to $d'(i,j)^{-r}$, with r a given real constant. To generate a proper probability distribution, the normalizing constant is given by $\sum_{v \in V} d'(i,v)^{-r}$. This probability distribution is called the *inverse r^{th}-power distribution.*

If p and q are given as fixed constants, this network family is described only by parameter r.

Now, a message is to be sent within this network. The transmission model is as follows: We start with two arbitrary nodes in the network, source node s and target node t. The goal is to transmit the message from s to t with as few steps as possible. An algorithm is defined as *decentralized* if at any time-step the current message holder u has knowledge of only:

DA 1) the set of local contacts among all nodes (i.e. the underlying grid structure),
DA 2) the position, $P(t)$, of target node t on the grid, and
DA 3) the locations and long-range contacts of all nodes that have come in contact with the message.

Here, we just want to state the results of this approach. The proofs can be found in [354]. The first result is that there is only one possible parameter for r in a twodimensional grid where a decentralized algorithm is able to perform the transmission task in expected $O(\log n)$ steps. This efficiency is measured as the *expected delivery time*, i.e., the number of steps before the message reaches its target:

Theorem 6.5.1. *Navigability in Kleinberg Small-Worlds*
There is a decentralized algorithm $\mathcal{A}$ and a constant α, independent of n, so that when $r = 2$ and $p = q = 1$, the expected delivery time is at most $\alpha \cdot (\log n)^2$.

The next theorem shows that $r = 2$ is the only parameter for which the expected delivery time is polynomial in $O(\log n)$:

Theorem 6.5.2. *(a) Let $0 \leq r < 2$. There is a constant α_r, depending on p, q, r, but independent of n so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(2-r)/3}$.*
(b) Let $r > 2$. There is a constant α_r, depending on p, q, r, but independent of n, so that the expected delivery time of any decentralized algorithm is at least $\alpha_r n^{(r-2)/(r-1)}$.

These results can be generalized for multi-dimensional spaces. For any k-dimensional space, a decentralized algorithm can construct paths of length polynomial in $O(\log n)$ if and only if $r = k$.

What does this decentralized algorithm look like? In each step, the current message-holder u chooses a contact that is as close to the target as possible, in terms of lattice distance. And that is all. Note, that this very simple algorithm does not make use of DA 3). Accordingly, we do not need any memorization of the route a message has taken to get to node i.

Summarizing, Kleinberg-small-worlds provide a way of building an overlay network for Peer-to-Peer applications, in which a very simple, greedy and local routing protocol is applicable. On the other hand, it requires some

information that is not naturally given in a Peer-to-Peer system, namely a distinct mapping for nodes and files to a k-dimensional position. In principle, this can be provided by a *distributed hash table* (DHT) approach but this is not always possible. For more information on DHTs see Chapter 7.

Second, we need a metric that allows us to measure the distance between two positions in the system. Third, and somewhat counter-intuitive to the decentralized approach, we need some global information, especially a list of **all** neighbors within a given distance and a list of all other nodes in the system and their distance to a given node, to choose the q longe range contacts. The next subsections give some approaches that try to achieve this. Protocols discussed in these subsections are explained in more detail in Chapter 8.

6.5.2 Small-World Overlay Networks in Peer-to-Peer Systems

Some papers indicate that Peer-to-Peer systems sometimes voluntarily evolve into a small-world [16, 302, 639]. For Freenet it could be shown that a low to medium load, in terms of the number of files in the system, leads to a small-world network. This is achieved by the following routing table update: Every file is correlated with a key, e.g., by a hash function. The file is originally stored at some node with a similar key. Each request is at every time forwarded to the one node in the routing table that has the closest key to the requested key. The request has a *time to live (TTL)*, i.e., there is a counter in the request that is incremented with every forwarding, and the request is removed when the counter reaches TTL. If a node has no more neighbors to route a request to, it will send a backtracking 'request failed' message. If the request is successful, the file will be sent over the routing nodes back to the requesting node. Every routing node will thereby save the file and add the sending node's key to its routing table. If either the file space or the routing table space is full, the *least recently used (LRU)* entry is replaced by the new entry. With this simple LRU-replacement algorithm, the system copies frequently requested files and most files in the file space are requested many times before they are replaced. But with a high load, i.e., a high number of different files and requests, the set of files at each node is rapidly changing and the number of successful requests for any of the stored files decreases. This unexpected behavior motivated Zhang, Goel, and Govindan to use a small-world overlay network to improve Freenet's performance [639]. The authors try to build a network in which most files on one node were 'close neighbors', and only some of the stored files are 'distant' files. The notion of distance $d(i, x)$ used here is given by the hashkey of each file i to a given random seed x from the key space S. The algorithm works as follows:

1. Each node i chooses a seed $s(x)$ randomly from the key space S when it joins the system.

2. When the datastore at a node is full and a new file f with key $key(f)$ arrives (from either a new insertion of a file or a successful request), the node finds out from the current datastore the file with key v farthest from the seed in terms of the distance in the key space S:

$$d_{\max}(datastore) = \max_{\text{file } g \text{ in datastore}} d(key(g), x) \tag{6.16}$$

(a) If $d(key(f), s(x)) < d_{\max}$ cache f and evict v. Create an entry for f in the routing table. This has the effect of clustering the keys in the routing table around the seed of the node.
(b) Otherwise, cache f, evict v, and create an entry for f in the routing table only with a probability p (randomness). This has the effect of creating a few random shortcuts.

The authors fixed p to 0.03 because this value worked best in the experiments conducted. The procedure is called the *Enhanced-clustering Cache Replacement Scheme* and produces routing table entries that resemble small-worlds in the sense that each node preferentially stores those files that are near to its own key. The authors show that this small, local improvement is able to increase the hit ratio significantly, where the *hit ratio* is defined as the ratio of the number of successful requests to the total number of requests made. The approach of Zhang, Goel and Govindan follows quite closely the original small-world concept of Watts and Strogatz.

Another approach that is more closely related to the Kleinberg small-worlds is given by a protocol named *Symphony* [400]. This approach is similar to the first in that it also relies on a hash-function that assigns each file a unique key with which it is addressable. The idea is that all nodes are placed on a circle with unit perimeter and every node is responsible for (stores) all files with a key equal to or greater than its own key and smaller than the key of the next node. This part of the circle is its *segment of responsibility*. The joining node draws its position on the circle uniformly at random from the interval $[0, 1[$ and connects to its next neighbor on each side. In this property it resembles the *Chord* protocol, introduced in [575]. Additionally, every node tries to connect to k randomly drawn nodes. The probability that a connection with a node responsible for x is established is given by the following probability distribution:

$$P(X == x) = \frac{1}{x \log n} \tag{6.17}$$

This approach has the problem that the total number of nodes n has to be known in advance. The authors estimate this value from the length of the circle each node is responsible for: If all nodes draw their position uniformly at random, the expected mean length of the segment of responsibility is $1/n$. Averaging these lengths over a set of known nodes, the number of nodes in the system can be approximated. The probability distribution belongs to

the family of harmonic distributions (a fact which inspired the name for the protocol). As in the Kleinberg paper, the actual routing protocol is greedy: Every message holder sends it to the one node known to have a key next to the requested file key.

The authors ensure that no node has more than a fixed number k of (incoming) long range contacts. If, by chance, one node asks to establish a long-range link to a node that has already reached this number, the latter will refuse the new connection. The most interesting property of this protocol is that it shows a trade-off between the number of links a node has and the expected path length within the network to find a file:

Theorem 6.5.3. *Symphony*
The expected path length in an n-node network with $k = O(1)$ edges, built by the Symphony protocol, is inversely proportional to k and proportional to $(\log n)^2$.

This is true whether long-range links are used in one direction only (from the one building it to the one randomly chosen) or in both directions.

The Symphony approach is elegant and smoothly transforms the idea of Kleinberg small-worlds to the world of Peer-to-Peer systems. An even more sophisticated approach was given by Hui, Lui and Yau in [309]. In their *Small-World Overlay Protocol (SWOP)*, clusters emerge in a self-organized way. The basic idea is again based on a hash-function and nodes that are placed on a unit-perimeter circle. Additionally, every node tries to connect to one random node with the probability distribution in Equation 6.17. Here, n is the number of clusters in the system.
A new node joining the circle will be the basis for a new cluster if both of its neighbors are members of a cluster with a maximum size. Otherwise, it will join the cluster with smaller size and create some *intra-cluster connections*. The maximal cluster size is given as a variable of the system and might be changed dynamically. Each cluster has one designated *head node* that is chosen by some periodically repeated voting mechanism. This head node is responsible for maintaining some 'long-range' inter-cluster connections. The routing protocol is the same greedy protocol used in the other approach: Each message holder will send the message to the one node known to have a key next to the requested file key.
The article is mainly concerned with the proper behavior of a protocol in a *flash crowd scenario*: These are situations, in which some static or dynamic object is heavily requested. The example provided by the authors is the crush on the CNN web server for news documents that was initiated by the 9/11 incident. Here, the news consists not only of static documents but might also be changing within minutes. A careful distribution within the net can prevent server crashes.
The idea proposed by the authors is that heavily requested documents should be copied via the *inter*-cluster links so that nearly every cluster has its own copy. This is sufficient in static scenarios, but an additional version number

has to be maintained if the document is changing. The source node can then send update messages over the long-range connections to other nodes holding the copy. The nodes within a cluster can be informed by using local links. Because of the small-world character of the protocol, the clustering coefficient of such a system is high and, depending on the cluster size and the actual value of the clustering coefficient, the update message soon reaches all members of the cluster.

6.5.3 Scale-Free Overlay Networks in Peer-to-Peer Systems

The most prominent feature of Scale-Free Networks is its fault tolerance [21, 129]. Since the degree is very heterogeneously distributed in the system, a random failure will very likely strike one of the nodes with low degree. These are most often not crucial for the connectivity of the network. It has been shown [21] that the diameter of the Internet at the autonomous system level in July, 2000 would not be changed considerably if up to 2.5% of the routers were removed randomly. This is an order of magnitude larger than the failure rate.

The authors further compared the fragmentation of random graphs and Scale-Free Networks. By randomly removing nodes from a random graph the network will soon fragment. For a failure rate of 5% in networks with 10,000 nodes the biggest connected component holds approximately 9,000 nodes. For a failure rate of 18% there is no biggest connected component any more, but only components with a size between 1 and 100. For a failure rate of 45% all components have only one or two nodes. For Scale-Free Networks the story is different: For a failure rate of 5% only some one- or two-node components break off the network. For a failure rate of 18%, the biggest connected component still holds 8,000 nodes with isolated clusters of size 1 to 5. Even for an unrealistically high failure rate of 45% the large cluster persists and the size of the broken-off fragments is below 12.

This behavior is desired for most Peer-to-Peer systems because it stabilizes the network structures in these highly dynamic systems. Fortunately, some of the protocols in Peer-to-Peer systems generate this favoured network topology for free: The idea of Gnutella is that every new node joining the system first connects to a handful of known servers. Later, it remembers some of the nodes involved in queries it is interested in. It could be shown that this behavior leads to pure scale-free or scale-free-like networks [516, 534].

On the other hand, the same scale-free architecture makes a network extremely vulnerable to attacks [21, 129, 532]: If it is possible to detect the hubs of the system and to attack them, e.g., with a Denial-of-Service-Attack, the network is more rapidly fragmented than the corresponding random graph. This led Keyani, Larson and Senthil to the idea of changing the network architecture as soon as an attack is detected. With a local protocol, each

node is enabled to decide whether the loss of a neighbor node is probably the result of an attack. To do so, the node periodically tries to connect to its immediate and second degree neighbors (neighbors of neighbors). If, in time period T, the percentage of lost second degree neighbors is greater than the percentage of lost direct neighbors and a given threshold P, an attack is assumed. For this eventuality, every node holds a list of random contacts to other nodes. This list is collected during normal work, e.g., while receiving a query or other messages. In case of a detected attack all detecting nodes establish connections to these nodes, and a random graph is generated. Of course, for 'friendly' times the more robust scale-free network is still favoured and will be restored after the attack is over.

6.6 Summary

This chapter has presented three prominent network models that are able to model different aspects of many complex and dynamic networks. First was the random graph model. It is easy to simulate and many properties can be analyzed with stochastic methods. It can be a good model for some Peer-to-Peer systems. Other Peer-to-Peer systems exhibit the so-called small-world effect: High clustering of nodes that share similar interests and just a few links between nodes with very different interests. These few 'long-range' or 'short-cut' links decrease the diameter such that the average path length in these networks is almost as short as in a random graph with the same number of nodes and links. Finally, we presented a model that generates scale-free networks. In these networks the presence of highly connected nodes ('hubs') is much more probable than in random graphs, i.e., the probability of finding a node $P(k)$ is proportional to $k^{-\gamma}$, where γ is a constant.

Small-world networks are interesting for Peer-to-Peer systems because they provide a good way to structure nodes with similar interests into groups without losing the small diameter of random graphs. Scale-free networks exhibit a good fault tolerance, but on the other hand, the are extremely vulnerable to attacks.

As shown, some authors have already tackled the problem of how desired properties of these three network models can be transmitted to overlay networks in Peer-to-Peer systems using simple and local protocols. Future research will have to show which kind of network model is best for building structured, yet self-organizing overlay networks for Peer-to-Peer systems that are stable despite dynamic changes and scale nicely under the constantly increasing number of peers.