# Peer-to-Peer Applications Beyond File Sharing: Overlay Network Requirements and Solutions

Vasilios Darlagiannis Oliver Heckmann Ralf Steinmetz

Technische Universität Darmstadt
Multimedia Communications (KOM)
Merckstr. 25, D-64283 Darmstadt, Germany
{Vasilios.Darlagiannis, Oliver.Heckmann, Ralf.Steinmetz}@KOM.tu-darmstadt.de

## Abstract

*In this paper we analyze the functional and non-functional requirements of Peer-to-Peer (P2P) systems that go beyond the needs of the well explored file-sharing P2P systems. Four basic subcategories are suggested to classify the non-functional requirements: Adaptability, Efficiency, Validity and Trust. Similarly, the functional requirements are divided in user-triggered and system-triggered functions. Then, we present and discuss several existing solutions following different design approaches in order to reveal their suitability for different types of P2P applications. More particularly, we capture the characteristics of structured and unstructured overlay networks. For the structured solutions, we investigate Distributed Hash Tables and lexicographic graph based solutions. For the unstructured approaches, we discuss both hierarchical and non-hierarchical overlay networks and emphasize on small-world and power-law networks. Finally, we provide a set of demanding application classes such as Voice over IP and Massive Multiplayer Online Games that may be benefited by utilizing the P2P technology.*

## 1 Introduction

Peer-to-Peer (P2P) systems received a significant attention by the networking community the last years. The P2P paradigm releases the need for dedicated servers that provide the required functionality and reduces the operation cost of many systems. Although P2P systems are mostly known from file-sharing application, a number of further applications are redesigned based on this paradigm. In this paper, we provide a survey of the functional and non-functional requirements and the available solutions for applications beyond the well explored file-sharing systems. A number of these applications is provided, though their number and application areas grow up continuously.

A definition of a P2P system is given in [73] where it is stated that a P2P system is a self-organizing system consisting of end-points (called "peers") forming an overlay network with a number of properties. Peers offer and consume services and resources and have significant autonomy. Services are exchanged between any participating peers. Long-term connectivity of individual peers cannot be assumed in a P2P system. This means that a P2P system has to explicitly deal with dial-up users, variable IP addresses, firewalls, NATs and that the system typically operates outside the domain name system.

The rest of this paper is organized as follows. In Section 2 the basic graph properties and definitions are provided and associated with P2P systems. In Section 3, the non-functional requirements are identified, followed by the functional ones in Section 4. The investigation of several available overlay network solutions is given in Section 5. The applications we have examined are then shortly described in Section 6. The paper concludes and provides a short outlook in Section 7.

## 2 Graph Properties and P2P Overlay Networks

Graph theory concepts are widely used to evaluate the static (and in certain cases the dynamic) properties of communication networks. It is necessary to supply the crucial graph theory term definitions, before studying the relevant graph models of several suggested P2P overlay networks. Further knowledge on the topic may be gathered from graph theory textbooks such as [43].

A *graph G* is a pair $G = (V, E)$, where the elements of $V$ are the *vertices* and the elements of $E$ are the *edges*. Throughout this paper, the terms *node* or *peer* are used alternatively to describe a vertice and the terms *connection* or *link* are also used to describe an edge. Edges can be *directed* (uni-directional) or *undirected* (bidirectional). A graph in which each edge is a directed edge is also called a *digraph*. For any vertex $u$ of a graph, the *degree* $(deg\ u)$ of $u$ is the number of edges attached to $u$. A graph is said to be *regular* of degree $r$ if all local degrees are of the same number $r$. The *order* or the *size* $|V|$ of a graph $V$ is the population of its vertices. The length $max_{u,v}d(u, v)$ of the "longest shortest path" between any two graph vertices $(u, v)$ of a graph is called the *diameter* of the graph.

Digraphs have been extensively used in interconnection networks for parallel and distributed systems design. Digraphs received special attention from the research community aiming to solve the problem of the so-called $(k, D)$ digraph problem. The goal is to maximize the number of vertices $N$ in a digraph of maximum out-degree $k$ and diameter $D$ [29]. Some general bounds relating the order, the degree and the diameter of a graph are provided by the well-known Moore bound [19]. Assume a graph with node degree $k$ and diameter $D$; then the maximum number of nodes (*graph order*) that may populate this graph is given by Equation 1:

$$N \leq 1 + k + k^2 + \ldots + k^D = \frac{k^{D+1} - 1}{k - 1}. \tag{1}$$

Interestingly, the Moore bound is not achievable for any non-trivial graph [19]. Nevertheless, in the context of P2P networks, it is more useful to reformulate Equation 1 in a way that provides a lower bound for the graph *diameter* $(D_M)$, given the node degree and the graph order [28]:

$$D_M = \lceil log_k(N(k-1)+1) \rceil - 1 \leq D. \tag{2}$$

Two particular classes of graphs have been adapted by several researchers for overlay network design: The well-known *small-world* graphs and the *scale-free* graphs. Their properties fit well with the effective operation of loosely structured overlay networks. An excellent survey on the issues of aforementioned graphs can be found in [77]. Moreover, lexicographic (i.e., de Bruijn) graphs [26] have been found to have very promising properties. Further, a $d$-regular graph is a $(d, c)$-expander or has a $c$-expansion (for some positive $c$) if and only if for every subset $U \subset V$ of size at most $|V|/2$. Expander graphs have been investigated also for P2P overlay networks [32] and [48], particularly focusing on their ability to expand exponentially by increasing linearly their diameter.

The graph properties are very important in evaluating the performance of the P2P overlay networks. They are closely related to certain requirements as they analyzed in Section 3, e.g., scalability, fault-tolerance etc.

## 3 Non-functional Requirements

The non-functional requirements are the most crucial criteria to decide whether an application can perform effectively. Therefore, we begin our survey by identifying a set of them that is mostly related to Quality of Service (QoS) as well as security in general. QoS is the well-defined behavior of a system with respect to certain parameters [69]. Between many of the properties there are clear trade-offs as improving one property often comes on cost of another. The concepts described in the following sub-sections are illustrated in Figure 1.

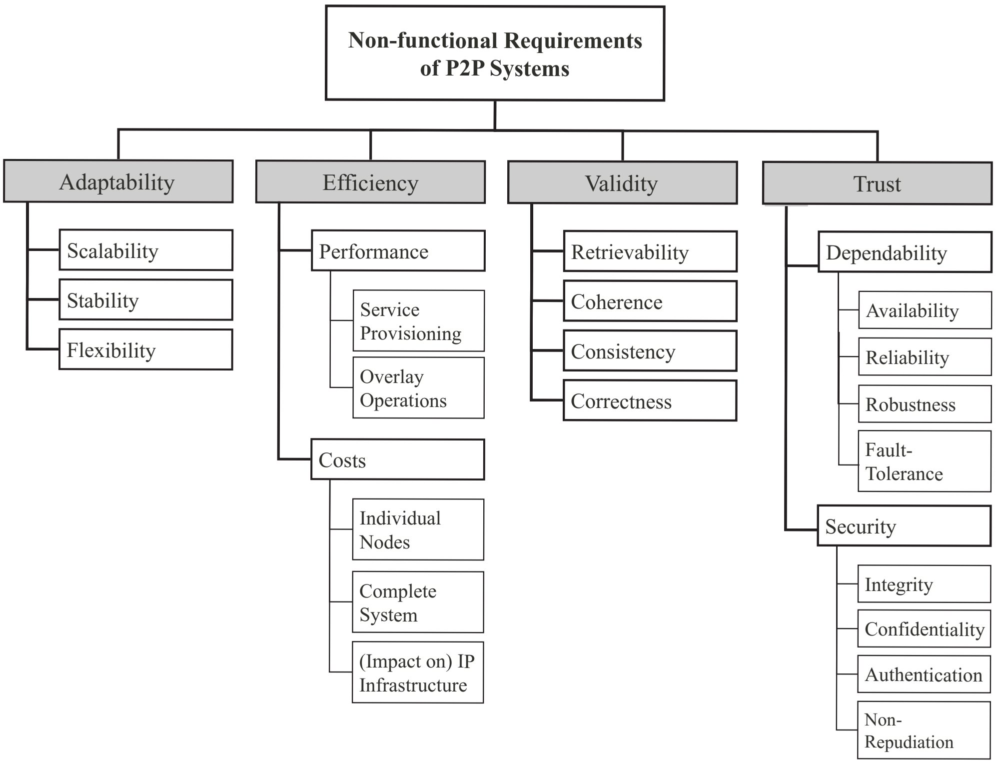


**Figure 1. Non-functional requirements for P2P systems.**

### 3.1 Efficiency

Efficiency describes the relationship of the performance of the P2P system and the costs that it incurs. Cost and performance are complex metrics with many facets that have to be taken into account. Costs can be classified as monetary and non-monetary as well as internal and external costs. To obtain a comprehensive picture of the costs incurred by the systems, one normally has to determine the different costs from different points of view like the total costs of the system (system point of view) and how these costs are distributed over the individual peers (load balancing and point of view of individual peers). It should also be taken into account that P2P systems can cause external costs to other applications and systems that share the same resources such as the connectivity bandwidth. For example, due to the tremendous amount of traffic generated by P2P file-sharing applications and their greedy downloading behavior, these P2P systems often significantly decrease the utility of other applications like e.g. realtime streaming applications. Performance also has different facets. One can generally distinguish between the performance of the overlay operations themselves (see Section 4) and the performance of the services running on the P2P network themselves (e.g. like file exchange or media streaming).

### 3.2 Adaptability

Under adaptability we subsume three related terms: (i) scalability, (ii) stability, and (iii) flexibility.

Scalability describes how well the system adapts to quantitative changes in several dimensions. The dimension can be the (i) the numerical dimension, which consists of the number of users, objects and services encompassed, (ii) the geographical dimension, which consists of the distance over which the system is scattered and (iii) the administrative dimension, which consists of the number of organizations that exert control over parts of the system [59]. Modern P2P systems are challenging systems that may scale enormously over all of the three identified dimensions. Peers may be distributed globally and each user may have the absolute control of each own machine. Typical sizes of P2P systems may reach several millions of users and it is envisaged to be extended to billions.

Stability describes the ability of the system to stabilize itself after changes due to “normal” system dynamics. The most frequent type of change in P2P networks is the joining and leaving of nodes. Stability is closely related to robustness (see Section 3.3).

The flexibility of a P2P system describes the qualitative adaptability of a system, that means how well it is suited for different applications and environments. Specialized systems will typically be less flexible but often more efficient than general systems or P2P frameworks like e.g. JXTA.

### 3.3 Trust

The level of trust that a user can have in a system depends on the systems security properties and its dependability.

Security is an umbrella term for many different properties. For P2P systems, probably data integrity, confidentiality, authentication, non-repudiation, and in some cases anonymity are the most important characteristics. Data integrity is the protection of the data in the system against accidental or malicious modification or destruction. Confidentiality has been defined by the ISO as "ensuring that information is accessible only to those authorized to have access". Authentication ensures that peers are who they say they are. Non-repudiation describes that a transaction cannot later be denied by one of the parties involved. Anonymity can be important for some P2P applications; it describes that the personal identity of a user or peer remains unknown [70].

Dependability subsumes availability, reliability, robustness, and fault-tolerance. A system has a high availability if it is ready to deliver the correct services at any time. It is reliable, if the services are continuously delivered correctly as specified. Robustness and fault-tolerance describe the ability to continue operation under or recover from severe system dynamics (like peak loads or major disruptions in network connectivity) and errors.

### 3.4 Validity

Data in P2P systems is widely distributed over many different peers with different users and can be modified by many of them at the same time. This raises the question of how valid the information and data are. To analyze validity, we look at the retrievability, coherence, consistency and correctness. Please note that different computer science communities use these terms in different ways.

For the purpose of this paper, the retrievability describes how well information/data stored in the P2P system can be found and retrieved by other peers. Investigating the retrievability means for example specifying the consequences of a search or lookup that yielded no results. Some P2P systems can guarantee that this means that (at the point in time the search was done) there was no matching data in the complete system while other (typically unstructured) P2P systems cannot give this guarantee as the completeness of their search results is not deterministic.

Coherence describes the freshness and timeliness the the search result, that means whether it contains the latest version of the data. Consistency describes whether all replica of one piece of data in the system are equal. Finally, the correctness of the data specifies whether objects stored in the P2P system fulfill integrity constraints according to their specification and semantic

## 4 Functional Requirements

Overlay operations may be divided in two categories, based on whether they are user-triggered functionality or system-triggered functionality. A *user* may not necessarily be a human being, but instead may be a *bot* or an *agent* or an advanced service or application that is designed to use system's functionality. System-triggered functionality consists of operations that aim at achieving efficient system operation. Figure 2 summarizes the common functionality. More advanced functionality such as application layer multicasting, anycasting or mobility may be offered by higher infrastructure levels.

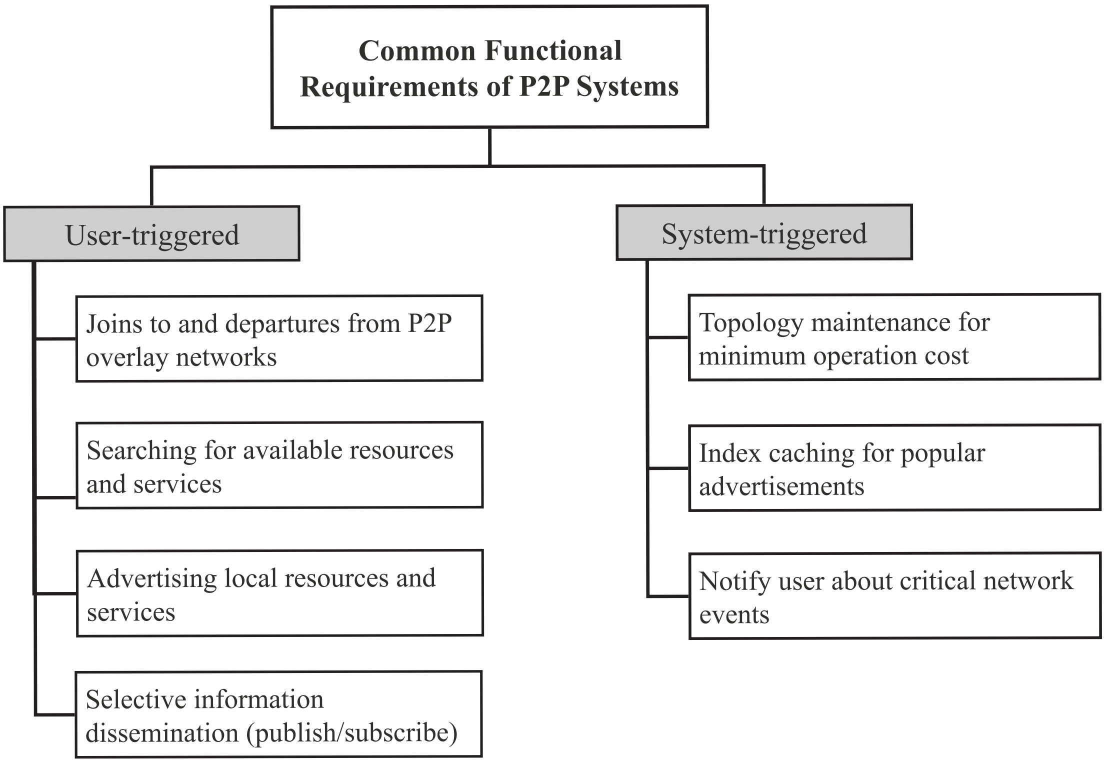


**Figure 2. Common functional requirements for P2P systems.**

### 4.1 User-triggered Functionality

As a first step, a peer must **join** the targeted P2P system. A bootstrap phase provides the necessary *"hooks"* to initiate the joining procedure. Common approaches met in widely deployed P2P systems include *history-based* mechanisms, *word-of-mouth*, well-known public Web sites and well-known *seed* peers. Another alternative for particular cases may be the multicast-based mechanism suggested in MPEG-4 DMIF [42], where users disseminate their existence in well-known multicast channels. Assuming that a peer has successfully obtained a valid address of another peer that is currently member of the system, the second phase of joining begins, where the new peer may be randomly or deterministically positioned in the overlay network.

While the join procedure is completed after a number of message-exchanging steps, in contrast, peers may **depart** arbitrarily from the system due to their complete autonomy. It may be advantageous for the system to provide incentives for peers to complete any pending operations and notify their neighbors about their impending departure before it occurs. However, such mechanisms are not typically met in most deployed systems.

The most important operation provided by a P2P overlay network is its ability to **search** for advertised items despite the large size of the system. However, the semantics of the search operation may differ resulting to various realizations. For example, file sharing systems (e.g., Gnutella) usually offer a text-based search operation where the requestor floods the network with messages that include the requested file name (or keywords) [23]. Then, it may receive back replies including advertisements, which have indexing information that matches the query text. However, such an approach may be very costly in terms of required communication load. In cases where an advertisement may be uniquely identified (e.g., using a "perfect" hash function), more advanced search mechanisms may be deployed. Such search operations are termed as **lookup** operations [12]. The efficiency of lookup operations is limited though in cases where the requested advertisement may be uniquely identified. This shortcoming motivated researchers to investigate advanced mechanisms to provide richer functionality for lookup operations. Such operations are called *complex queries* [35] or *range queries* [8]. SHARK [58] is such a system, which capitalizes on grouping the users based on their interests. Moreover, Heckmann et al. [37] describes an overlay network optimized or location-based area search.

Nevertheless, before peers are enabled to search for useful advertisements, local peer resources and services must be **advertised** first. The advertisements may be stored locally or remotely (or both). The advertised information may be as simple as a text field or a *globally unique identifier* (GUID) or even rich metadata structures. The indexing information may hold the complete information or some aggregated summary.

Another useful operation is the ability to selectively disseminate information, the so-called **publish/subscribe** systems [9]. Similarly, users may find useful to be asynchronously informed when important events occur in the system.

### 4.2 System-triggered Functionality

System-oriented operations aim at maintaining a useful state in the system so that user-oriented operations may be more efficiently performed. The most important system-oriented operation is the **maintenance** of the targeted overlay network topology in order to utilize structure-related benefits. However, proactive mechanisms (mechanisms, which operate in advance targeting to predict the future state, thus, improving system performance) that are frequently used in maintaining the perfect topology may cause significant amounts of traffic in scenarios where peers join and leave very frequently.

Moreover, **caching** (though an optional operation) may help reducing the operational cost of the system. Apparently, caching is suggested in several distributed system technologies (e.g., Web technologies [10], or distributed file systems such as AFS [40]). Appropriate caching policies that consider the intrinsic characteristics of P2P systems have been proven very useful, especially in scenarios where query distributions and resource popularity do not follow uniform distributions.

For many cases, it is useful to **notify** the user level to get a decision on a critical event, which are of high importance for the user. For example, the user may be informed when particular replies to posed queries are received in order to select further operations.

## 5 Overlay Network Solutions

A vital abstraction of P2P systems is the employed overlay network topology, which is supplied by embedded protocols, mechanisms and algorithms. Several approaches have been proposed as communication schemes in order to provide efficient and scalable inter-peer communication. These schemes are designed on top of the physical networking infrastructure as overlay networks. Their topologies and operating mechanisms influence greatly the performance of routing and topology maintenance algorithms and hence, the efficiency of the involved P2P system.

## 5.1 Structured Approaches

Structured overlay networks tightly control their topology and place the indexing information for the advertised resources at specified locations. Thus, subsequent queries can be routed to these locations and lookup can become more efficient.

### 5.1.1 Distributed Hash Tables

The great majority of modern structured P2P systems use *Distributed Hash Tables* (DHTs) as a communication infrastructure[1]. DHTs are powerful abstractions, where indexing information is placed deterministically at the corresponding peers with a GUID $(N_{id})$ closest to the data object's advertisement unique key $(R_{id})$. DHTs use the defined hash function to select the way resources should be treated. In fact, using widely acceptable hash functions in distributed environments is a way to provide a communication mechanism without the need to exchange messages. DHTs provide a scalable way to store and retrieve data objects under given keys [38]. Each key lookup is resolved in multiple steps, resulting in a multi-hop path to be taken in the overlay. Several DHT-based overlay networks are available in the literature.

*Chord* [74] is one the first DHT-based approaches, which utilizes the *consistent hashing* [46] mechanism to assign keys to its peers. Consistent hashing is designed to let peers enter and leave the network with minimal interruption. This decentralized scheme tends to balance the load on the system, since each peer receives roughly the same number of keys, and there is little movement of keys when peers join and leave the system. In the steady state, each peer maintains routing state information for $O(logN)$ other peers, where $N$ is the population of the network. Node identifiers (i.e., $N_{id_i}$ for node $i$) are ordered on an identifier circle using a modulo operation (with operand $2^m$), thus holding that $0 \leq N_{id_i} \leq 2^m - 1$, for each peer. In this scheme, key $R_{id} = k$ is assigned to the peer whose $N_{id_i}$ is equal to or immediately follows $k$ in the identifier space (assuming that there is no peer $j$ in the system so that $k < N_{id_j} < N_{id_i}$).

*Pastry* [67] and *Tapestry* [79] make use of Plaxton-like prefix routing, to build a self-organizing de-

[1]Early approaches of distributed systems developed distributed structures based on Linear Hashing [50].

centralized overlay network. Plaxton et al. [61] proposes a distributed data structure, known as the *"Plaxton mesh"*, optimized to support a network overlay for locating named data objects which are connected to one root peer. Pastry and Tapestry employ decentralized randomness to achieve both load distribution and routing locality. However, Tapestry uses a suffix-based routing mechanism while Pastry a prefix-based one. Moreover, the handling of network locality and data object replication is performed in a different way. The architecture of Tapestry improves the Plaxton mesh structure with additional mechanisms to provide availability, scalability, and adaptation in the presence of failures and attacks.

*Kademlia* [56] is a symmetrical DHT-based overlay that uses a XOR-based distance metric to construct its topology and assign the resource advertisements to peers. Kademlia's symmetrical architecture enables the usage of query messages for maintenance purposes, thus, reducing the required out-of-band maintenance signalling. Kademlia allows peers to select their neighbors from sets of peers sharing the same prefix. Kademlia, Pastry and Tapestry have operation complexity comparable to that achieved by Chord.

The *Content Addressable Network (CAN)* [64] is a distributed decentralized P2P infrastructure. The architectural design is a virtual multi-dimensional Cartesian coordinate space on a multi-torus. The entire coordinate space is dynamically partitioned among all the peers ($N$ number of peers) in the system such that every peer possesses its individual, distinct zone within the overall space. Each peer maintains $O(d)$ neighbors and the lookup procedure requires $O(d\sqrt[d]{N})$ steps.

*SkipNet* [36] and *SkipGraph* [11] are two very similar structured overlay networks (though they have been developed independently) that extend skip lists [63], a probabilistic data structure. While they are similar to Chord, their basic difference is that they release the requirement that fingers (shortcut connections) must be exponentially distributed. SkipNet and SkipGraph permit peers to have fingers that are randomly located shortcuts.

*Viceroy* [54] is a structured network based on the butterfly topology. Viceroy requires only a constant number of neighbors with high probability while its diameter is growing up logarithmically. Though, the construction and maintenance procedures are relatively complex.

As it can be observed, the design of P2P overlay networks attracted a great interest from the research community. The list can be extended (though not exhaustively) to include *AGILE*[57], *Kelips* [34], *P-Grid* [1], *HyperCup* [68], and *pSearch* [75]. Moreover, several interesting surveys provide comparisons among most of the well-known systems (cf. [7], [52]).

### 5.1.2 Lexicographic (de Bruijn) Digraphs

An interesting class of digraphs are the lexicographic digraphs, which includes the de Bruijn and Kautz digraphs [15][2]. de Bruijn graphs have asymptotically optimal graph diameter and average node distance [51]. For binary de Bruijn graphs (with fixed out-degree 2) the maximum number of nodes is always limited by $2^D$ [3]. The graph contains $2^{D+1}$ directed edges in this case. Each node is represented by $D$-length strings. In the general case, each node is represented by a string such as $u_1u_2...u_D$. The connections between the nodes follow a simple left shift operation from node $u_1(u_2...u_D)$ to node $(u_2...u_D)u_x$, where $u_x$ can take one of the possible values $(0, k-1)$.

de Bruijn graphs have been suggested to model the topology of several P2P systems, though. Considerable examples of P2P systems that use de Bruijn graphs are Koorde [44], Omicron [25] D2B [30] and Optimal Diameter Routing Infrastructure (ODRI) [51].

*Koorde* [44] is a proposal that deploys the Chord design over de Bruijn digraphs. The authors suggest the construction of de Bruijn digraphs with node degree proportional to the logarithmic size of the network to avoid the robustness limitations of constant degree connectivity. This requires a good estimation of the size of the network and it obligates the most attractive feature of the de Bruijn digraphs (which is the combination of having logarithmic diameter and constant node degree). Koorde suggests the introduction of "imaginary nodes" to address the incremental extendability limitation of the de Bruijn graphs.

*D2B* [31] is a content addressable network that employs de Bruijn graphs to construct its overlay network. Although the proposed topology is a variation of de Bruijn graphs, they provide an interesting graph operation analysis. In D2B a procedure is suggested that allows nodes to have variable length identifiers of more than one symbol. This is even the case for linked neighbors. The resulting digraph is not always a de Bruijn one.

*Omicron* [25] is a two-tier de Bruijn based overlay network that addresses many of the identified non-functional requirements. A graph construction algorithm based only on local information has been

[2] de Bruijn graphs are less dense than Kautz graphs but they are more flexible since they do not have any limitations on the sequence of the represented symbols in every node.

[3] The Moore bound determines always maximum upper bounds on the size of the graphs that are not reachable for non-trivial cases.

proposed to define de Bruijn variants with incremental expandability properties. Moreover, in order to address the fixed node degree (preferably as small as possible to keep the graph maintenance costs low) that poses resiliency concerns, Omicron introduces the concept of peer *clusters* that guarantee network stability. This *hybrid* topology provides a *tightly structured* network. In parallel, it gives the freedom of selecting neighbor peers from several members of the neighbor clusters. This selection can be driven by various policies and metrics, i.e., by the efficient mapping to the underlying network or by satisfying trust-level requirements. Additional mechanisms have been proposed for the intra-cluster organization that deals with peer heterogeneity. For this issue, a *role*-based approach has been investigated. The common, basic operations of P2P overlay networks have been identified assigning a role to each of them. More specifically, four core roles have been identified: *Routers, Cachers, Indexers and Maintainers*. Peers are assigned with roles based on their capabilities and their predicted behavior so that each peer can contribute in an efficient way without hindering and degrading the overall performance.

### 5.2 Unstructured Approaches

Loosely structured (or simply *unstructured*) overlay networks do not aim to reach a predefined targeted topology as their structured counterparts do, but rather they have a more “random” structure. However, it has been observed that certain connectivity policies (i.e., preferential attachment) may result in topologies described by power-law networks or networks with small-world characteristics. Unstructured topologies are typically inefficient in finding published, rare items and the embedded searching operations are in general costly in terms of network overhead (most approaches use flooding or at best, selective dissemination mechanisms [53]). Nevertheless, in scenarios where the query distribution is non-uniform (i.e., lognormal, Zipf) unstructured networks may operate efficiently.

Unstructured networks may be designed are non-hierarchical and hierarchical. Gnutella 0.4 [33] is an example of the former, which offers a non-hierarchical P2P approach with minimum maintenance cost. However, the employed *flooding* mechanism used for querying makes Gnutella unscalable [66] and caused a system breakdown in the end of 2000 when the number of users increased considerably.

Therefore, hierarchical approaches became more popular as solutions to deployed systems. The concept of *super-peers*, i.e., peers with additional capabilities, is introduced in these systems. Super-peers form usually the backbone of the overlay network having normal peers placed around them. The architectures of eDonkey [27] and KaZaA [49] may be considered typical representatives of this design direction. The basic drawback of this approach is the requirement for the existence of super-peers, which subsequently imposes new requirements on the system. Super-peers have to operate legitimately since their actions have greater effect in overlay operations. Further, super-peer failures are more severe than normal peer failures. Also, malicious behavior of super-peers has far greater impact compared to the behavior of peers in non-hierarchical P2P systems. Additionally, there is lack of incentives for super-peers to serve the rest of peers. Super-peers may also become performance bottlenecks if there is not a sufficient number of them.

### 5.2.1 Small-world Networks

From a graph theoretic point of view, a small world network is a graph exhibiting a high clustering coefficient, but low characteristic path length (diameter). More precisely, the *clustering coefficient* $CC$ of a graph is the metric to characterize a network as having the small-world property. It is defined as the average fraction of pairs of neighbors of a node that are also neighbors of each other. Suppose that a node $i$ in the network has $c_i$ neighbor nodes. Apparently, at most $c_i(c_i - 1)/2$ edges can exist among them, and this occurs when every neighbor of node $i$ is connected to every other neighbor of node $i$. The clustering coefficient $CC_i$ of node $i$ is then defined as the ratio between the number $E_i$ of edges that actually exist among these $c_i$ nodes and the total possible number:

$$CC_i = \frac{2E_i}{(c_i(c_i - 1))}. \tag{3}$$

The clustering coefficient $CC$ of the whole network is the average of $CC_i$s. Obviously, $CC \leq 1$ where the equality holds if and only if every node in the network connects to every other node. In a completely random network consisting of $N$ nodes, $CC \sim 1/N$ (which is very small as compared to most real networks). On the other hand certain structures of regular graphs have a high clustering coefficient. Typically, small world networks combine two characteristics that are attractive for P2P systems (i) relatively low diameter and (ii) high connectivity (resulting from the high clustering coefficient).

The properties of small-world networks have been exploited in the context of P2P networks by multiple approaches, e.g., the DIET network [39], the Swan network [18], the Symphony network [55], as well as in an improvement of the original Freenet network [78].

### 5.2.2 Scale-free Networks

The distribution of the node degree in a network is a significant factor for several network properties. A number of real life complex networks, e.g., income of individuals, genera, Internet file sizes [65], the World Wide Web, metabolic systems, paper co-authorship, movie actors [5], cognitive sciences [72], etc. appear to have a power law distribution of the form $P(k) = k^{-y}$ where for the most typical cases $1 \leq y \leq 3$ [77].

Apparently, for one of the most well-known P2P networks, the Gnutella network, it has been shown that its nodes follow a power law distribution (though not built implicitly into its design) [4], [45] [62], [60], [3]. The underlying mechanism for this evolution is the *preferential attachment* mechanism, where the probability $\Pi_i$ of connecting to peer $i$ with degree $c_i$ is $\Pi_i = c_i / \sum_j c_j$. Peers tend to connect to well-known peers with higher probability ending up in the so-called *"rich get richer"* phenomenon.

Networks that have power law distributed node degree are called *scale-free* networks. While such networks have desirable characteristics such as relatively small diameter and can effectively support heterogeneity, they are vulnerable to attacks and diseases dissemination [6], [13]. Moreover, a node with high degree holds an important position in the network. A possible removal of the node can drastically change properties such as the diameter and might result in multiple smaller graphs (*network fragmentation*). Moreover, if the graph is considered as a communication network, nodes with high degree are involved in delivering a large amount of traffic, ending up to potential traffic *"bottlenecks"*.

# 6 Peer-to-Peer Applications Beyond File Sharing

After analyzing the functional and non-functional requirements of demanding P2P applications and providing a comprehensive list of available overlay network approaches, we focus on applications that have been developed so far by a number of researchers and the solutions which have been selected to meet their requirements.

## 6.1 Voice over IP and Instant Messaging

*Voice over IP* (VoIP) [2] and *Instant Messaging* (IM) are increasingly popular communication means. While initially they were designed as centralized systems, lately a great effort has been noticed in redesigning them following the P2P paradigm.

A very successful example of VoIP and IM system based on P2P technologies is Skype [71]. Skype is based on the hierarchical unstructured overlay network of FastTrack protocol used by KaZaA [49]. In fact, this was one of the main practical reasons for the wide acceptance of Skype. The inability of standard IP-Telephony protocols (such as SIP or H.323) to dominate in complex network environments, while in contrast, FastTrack ability to operate smoothly even behind firewalls, is a major factor for the success of Skype. Moreover, the low cost solution of Skype led the market towards this solution [76].

Skype uses a login server to store user names and passwords that performs also user authentication. The login server is the only centralized point in the Skype architecture. Skype extends the KaZaA techniques with a "global index" mechanism to guarantee finding a user who has logged in the network the past 72 hours. A detailed description of the Skype protocol can be found at [14].

SOSIMPLE [20] is an effort to reuse existing clients from the open source community. SOSIMPLE combines the SIP/SIMPLE family of IETF standards for VoIP and IM with a DHT solution (Chord has been selected for the first prototype). SOSIMPLE releases the requirement for a central proxy server and investigates the related security issues, by focusing mostly at the authentication phase.

### 6.2 Collaborative Virtual Environments

*Collaborative Virtual Environments* (CVEs) and *Massive Multiplayer Games* (MMGs) are very demanding systems, since they integrate multiple media types with various demanding network requirements and they involve a large number of frequently interacting users. Several architectures have been proposed in the past to deal with the communication overhead and the management of the collaboration sessions. Basically, two major directions have been followed: (i) the traditional C/S paradigm and (ii) the all-to-all server-less approach. However, both of them have limitations in terms of scalability. C/S approaches require additional server resource capabilities, as the number of clients increases. All-to-all approaches have high bandwidth requirements in order to enable communication among the participants and even harder to solve problems, such as global information consistency. Further, additional issues have to be addressed, e.g., fault-tolerance or heterogeneity of the participants. Network-layer multicasting could (if available) improve the efficiency of the all-to-all approach.

Darlagianis et al. [24] applies and adapts Omicron in CVE system design. Though the addressed issues are limited to a certain type of information (i.e., the state of mutable objects) it can be expanded to efficiently address additional communication needs of these systems. Additionally, a significant research effort has been focusing lately in designing *Massive Multiplayer Games* (MMGs) based on P2P technology. The game industry seems to be an important driving factor for the future research on this field. In particular, Knutsson et al. [47] developed SimMud, an experimental MMG prototype on top of FreePastry (an open source P2P platform based on Pastry). FreeMMG [21] is an open source initiative to support MMGs. Also, a scalable publish/subscribe system for Internet games called Mercury [16] has been developed, and in parallel, Iimura et al. [41] developed a system to support multiplayer online games.

### 6.3 Further Applications

Several more application areas investigate the P2P paradigm in order to take advantage of its characteristics. In this subsection we provide some further applications, however, it is not intended to be complete, but rather to capture the application possibilities.

Cramer et al. [22] describe a fully decentralized P2P video recorder where every peer is capable of receiving and recording digital satellite TV. Like a normal video recorder, users can program their machines to record certain programmes. The P2P system assigns the different recording jobs to different peers. Users can also store past broadcasts. That way, the system serves as a short-term archival storage for TV programs.

Heckmann et al. [37] describe a novel P2P system that is optimized for supporting area searches based on the geographical position of users and services. It can be used for different applications like for example virtual window shopping, as restaurant finder, or as implemented by the authors to connect web-cams to a self-organizing P2P network.

The application of the P2P paradigm in supporting distributed network management with a framework is investigated in [17]. Its overlay consists of several Distributed Network Agents managing networks fault-tolerance and performance. Kademlia is employed in the development of the related overlay network. The DHT mechanism implements a distributed index for the localization of other agents or resources.

## 7 Conclusions

In this survey paper, the non-functional requirements of P2P systems have been identified and defined comprehensively. A set of commonly seen user and system triggered functional requirements have been described to collect the necessary functionality. A number of solutions following structured or unstructured approaches are described and their advantages and disadvantages are discussed. As long as lookup based search fits to the search requirements, structured approaches can perform more efficiently (if peer churn is low). However, hierarchical approaches that borrow some client/server concepts dominate in several systems today. Finally, we list several application areas that are either novel or redesigned approaches that release the requirement for dedicated servers.

Research in P2P systems has so far focused mostly on a small subset of the non-function requirements of Figure 1, e.g. on scalability. It is important for future research projects to investigate the other requirements and their mutual interdependencies.

## References


[1] K. Aberer, P. Cudr-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, R. Schmidt, and J. Wu. Advanced Peer-to-Peer Networking: The P-Grid System and its Applications. *PIK - Praxis der Informationsverarbeitung und Kommunikation, Special Issue on P2P Systems*, 26(3):86–89, 2003.

[2] R. Ackermann. *Gateways and Components for Supplementary IP Telephony Services in Heterogeneous Environments*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, July 2003.

[3] L. A. Adamic, R. M. Lukose, and B. A. Huberman. *Handbook of Graphs and Networks: From the Genome to the Internet*, chapter Local Search in Unstructured Networks. Wiley, Berlin, 2002.

[4] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Physical Review E*, 64(046135), 2001.

[5] R. Albert and A.-L. Barabasi. Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, 74(47), 2002.

[6] D. Aldous. A Stochastic Complex Network Model. *Electronic Research Announcements of the Ammerican Mathematical Society*, 9:152–161, 2003.

[7] S. Androutselis-Theotokis and D. Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 36(4):335–371, 2004.

[8] A. Andrzejak and Z. Xu. Scalable, Efficient Range Queries for Grid Information Services. In *Proceedings of the Second IEEE International Conference on Peer-to-Peer Computing (P2P2002)*, September 2002.

[9] M. Antollini, M. Cilia, and A. Buchmann. Implementing a High Level Pub/Sub Layer for Enterprise Information Systems. In *Proceedings of the Conference on Enterprise Information Systems*, May 2005.

[10] M. Arlitt, R. Friedrich, and T. Jin. Performance evaluation of Web proxy cache replacement policies. *Performance Evaluation*, 39(1-4):149–164, 2000.

[11] J. Aspnes and G. Shah. Skip Graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, USA, 12–14 2003.

[12] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up Data in P2P Systems. *Communications of the ACM*, 46(2):43–48, 2003.

[13] A. Barabasi and E. Bonabeau. Scale-Free Networks. *Scientific American*, 288(5):60–69, 2003.

[14] S. A. Baset and H. Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Technical Report CUCS-039-04, Columbia University, September 2004.

[15] F. Bernabei, V. D. Simone, L. Gratta, and M. Listanti. Shuffle vs. Kautz/De Bruijn Logical Topologies for Multihop Networks: a Throughput Comparison. In *Proceedings of the International Broadband Communications*, pages 271–282, 1996.

[16] A. R. Bharambe, S. Rao, and S. Seshan. Mercury: A Scalable Publish-Subscribe System for Internet Games. In *ACM Netgames*, April 2002.

[17] A. Binzenhoefer and K. Tutschku. DNA: A Peer-to-Peer Framework for Distributed Network Management. In *Proceedings of Peer-to-Peer Systems and Applications Workshop associated with KiVS'05*, pages 135–138, March 2005.

[18] E. Bonsma. Fully Decentralized, Scalable Look-up in a Network of Peers using Small World Networks. In *Proceedings of Systemics, Cybernetics and Informatics (SCI)*, 2002.

[19] W. Bridges and S. Toueg. On the impossibility of directed Moore graphs. *Journal of Combinatorial Theory Series B*, 29:339–341, 1980.

[20] D. A. Bryan, B. B. Lowekamp, and C. Jennings. SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System. In *Proceedings of the 2005 International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA 2005)*, June 2005.

[21] F. Cecin, J. Barbosa, and C. Geyer. FreeMMG: An Hybrid Peer-to-Peer, Client-Server, and Distributed Massively Multiplayer Game Simulation Model. In *Proceedings of the 2nd Brazilian Workshop on Games and Digital Entertainment (WJogos'03)*, November 2003.

[22] C. Cramer, K. Kutzner, and T. Fuhrmann. Distributed Job Scheduling in a Peer-to-Peer Video Recording System. In *Proceedings of the Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications (PEPPA) at Informatik 2004*, September 2004.

[23] F. M. Cuenca-Acuna and T. D. Nguyen. Text-Based Content Search and Retrieval in ad hoc P2P Communities. In *Proceedings of International Workshop on Peer-to-Peer Computing*. Springer-Verlag, May 2002.

[24] V. Darlagiannis, A. Mauthe, N. Liebau, and R. Steinmetz. Distributed Maintenance of Mutable Information for Virtual Environments. In *Proceedings of the 3rd IEEE International Workshop on Haptic Audio Visual Environments and their Applications - HAVE'04*, pages 87–92, October 2004.

[25] V. Darlagiannis, A. Mauthe, and R. Steinmetz. Overlay Design Mechanisms for Heterogeneous, Large Scale, Dynamic P2P Systems. *Journal of Networks and System Management*, 12(3):371–395, 2004.

[26] N. G. de Bruijn. A combinatorial problem. In *Proceedings of the Koninklijke Academie van Wetenschappen*, pages 758–764, 1946.

[27] eDonkey2000. http://www.edonkey2000.com, 2005.

[28] M. Fiol and A. Llado. The Partial Line Digraph Technique in the Design of Large Interconnection Networks. *IEEE Transactions on Computers*, 41(7):848–857, 1992.

[29] M. Fiol, L. A. Yebra, and I. A. de Miquel. Line Digraph Iterations and the (d,k) Digraph Problem. *IEEE Transactions on Computers*, 33(5):400–403, 1984.

[30] P. Fraigniaud and P. Gauron. An Overview of the Content-Addressable Network D2B. In *Annual ACM Symposium on Principles of Distributed Computing*, July 2003.

[31] P. Fraigniaud and P. Gauron. The Content-Addressable Network D2B. Technical Report 1349, LRI, Univ. Paris-Sud, Paris, France, January 2003.

[32] C. Gkantsidis, M. Mihail, and A. Saberi. Random Walks in Peer-to-Peer Networks. In *Proceedings of IEEE INFOCOM 2004*, March 2004.

[33] Gnutella. http://www.gnutella.com, 2005.

[34] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse. Kelips: Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03)*, February 2003.

[35] M. Harren, J. Hellerstein, R. Huebsch, B. T. Loo, S. Shenker, and I. Stoica. Complex queries in DHT-based peer-to-peer networks. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02)*, 2002.

[36] N. J. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. SkipNet: A Scalable Overlay Network with Practical Locality Properties. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS '03)*, March 2003.

[37] O. Heckmann, M. G. Sanchis, N. Liebau, and R. Steinmetz. A Peer-to-Peer System for Location-based Services. In *Under submission*, 2006.

[38] K. Hildrum, J. D. Kubiatowicz, S. Rao, and B. Y. Zhao. Distributed Object Location in a Dynamic Network. In *Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, pages 41–52. ACM Press, 2002.

[39] C. Hoile, F. Wang, E. Bonsma, and P. Marrow. Core specification and experiments in DIET: a decentralised ecosystem-inspired mobile agent system. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 623–630. ACM Press, 2002.

[40] J. Howard. An Overview of the Andrew File System. In *Proceedings of the USENIX Winter Technical Conference*, pages 23–26, January 2004.

[41] T. Iimura, H. Hazeyama, and Y. Kadobayashi. Zoned Federation of Game Servers: a Peer-to-Peer Approach to Scalable Multi-player Online Games. In *Proceedings of the NetGames, ACM SIGCOMM'04 Workshops*, August 2004.

[42] ISO/IEC. 14496-6: Information Technology – Coding of audio-visual objects, Part 6: Delivery Multimedia Integration Framework, 1999.

[43] D. Jungnickel. *Graphs, Networks and Algorithms*. Springer, 1st edition, 1999.

[44] F. Kaashoek and D. R. Karger. Koorde: A Simple Degree-optimal Hash Table. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03)*, February 2003.

[45] M. Kabanov. In Defense of Gnutella. Online-Artikel, http://www.gnutellameter.com/gnutella-editor.html, 2001.

[46] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 654–663. ACM Press, 1997.

[47] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-Peer Support for Massively Multiplayer Games. In *The 23nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, March 2004.

[48] C. Law and K.-Y. Siu. Distributed Construction of Random Expander Networks. In *Proceedings of IEEE INFOCOM 2003*, April 2003.

[49] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the KaZaa Network. In *3rd IEEE Workshop on Internet Applications (WIAPP'03)*, June 2003.

[50] W. Litwin, M.-A. Neimat, and D. A. Schneider. LH*?a scalable, distributed data structure. *ACM Transactions on Database Systems (TODS)*, 21(4):480–525, 1996.

[51] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh. Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience. In *Proceedings of ACM SIGCOMM'03*, pages 395–406, August 2003.

[52] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Survey and Tutorial*, March 2004.

[53] Q. Lv, S. Ratnasamy, and S. Shenker. Can Heterogeneity Make Gnutella Scalable? In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02)*, March 2002.

[54] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: a Scalable and Dynamic Emulation of the Butterfly. In *Proceedings of the 21th Annual Symposium on Principles of Distributed Computing*, pages 183–192. ACM Press, 2002.

[55] G. S. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed Hashing in a Small World. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and System (USITS '03)*, March 2003.

[56] P. Maymounkov and D. Maziéres. Kademlia: A Peer-to-peer Information System Based on the XOR metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02)*, 2002.

[57] J. Mischke and B. Stiller. Peer-to-Peer Overlay Network Management Through AGILE. In *Kluwer Academic Publishers, IFIP/IEEE International Symposium on Integrated Network Management (IM)*, March 2003.

[58] J. Mischke and B. Stiller. Rich and Scalable Peer-to-Peer Search with SHARK. In *5th International Workshop on Active Middleware Services (AMS 2003)*, June 2003.

[59] B. C. Neuman. *Readings in Distributed Computing Systems*, chapter Scale in Distributed Systems, pages 463–489. IEEE Computer Society Press, 1994.

[60] W. H. Nicholas Gibbins. Scalability Issues for Query Routing Service Discovery. In *Proceedings of the Second Workshop on Infrastructure for Agents, MAS and Scalable MAS at the Fourth International Conference on Autonomous Agents (ICMAS2001)*, pages 209–217, 2001.

[61] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing Nearby Copies of Replicated Objects in a Distributed Environment. In *Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*, pages 311–320. ACM Press, 1997.

[62] M. Portmann, P. Sookavatana, S. Ardon, and A. Seneviratne. The cost of peer discovery and searching in the Gnutella peer-to-peer file sharing protocol. In *Proceedings of the International Conference on Networks*, pages 263–268, 2001.

[63] W. Pugh. Skip lists: a probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, 1990.

[64] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable Content Addressable Network. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 161–172. ACM Press, 2001.

[65] W. J. Reed and B. D. Hughes. From gene families and genera to incomes and internet file sizes: Why power laws are so common in nature. *Physical Review E*, 66(067103), December 2002.

[66] J. Ritter. Why Gnutella Can't Scale - No, Really. Online-Article, http://www.darkridge.com/jpr5/doc/gnutella.html, 2001.

[67] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[68] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. HyperCuP - Hypercubes, Ontologies and P2P Networks. In *Proceedings of the Agents and Peer-to-Peer Systems*, July 2002.

[69] J. Schmitt. *Heterogeneous Network Quality of Service Systems*. PhD thesis, Technische Universität Darmstadt, October 2000.

[70] B. Schneier. *Applied Cryptography*. John Wiley & Sons, Ltd, 2nd edition, 1996.

[71] Skype Website. http://www.skype.com/, 2004.

[72] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–425, 2004.

[73] R. Steinmetz and K. Wehrle. Peer-to-Peer-Networking and -Computing. *Informatik Spektrum, Aktuelles Schlagwort*, 27(1):51–54, 2004.

[74] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable Peer-to-Peer Lookup Service for Internet Applications. *IEEE Transactions on Networking*, 11(1):17–32, February 2003.

[75] C. Tang, Z. Xu, and M. Mahalingam. pSearch: Information Retrieval in Structured Overlays. In *Proceedings of the 1st Workshop on Hot Topics in Networks (HotNets-I)*, October 2002.

[76] A. Tapio. Future of telecommunication - Internet telephony operator Skype. Peer-to-peer technologies, networks and systems, Seminar on Internetworking, Spring 2005, 2005.

[77] X. F. Wang and G. Chen. Complex networks: Small-World, Scale-Free and Beyond. *IEEE Circuits and Systems Magazine*, 3(1):6–20, 2003.

[78] H. Zhang, A. Goel, and R. Govindan. Using the Small-World Model to Improve Freenet's Performance. In *Proceedings of IEEE Infocom*, 2002.

[79] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz. Tapestry: A Resilient Global-scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.