University of Würzburg
Institute of Computer Science
Research Report Series

On the Stochastic Scalability of Information Sharing Platforms

Phuoc Tran-Gia and Andreas Binzenhöfer

Report No. 364 June 2005

Department of Distributed Systems
Institute of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{trangia,binzenhoefer}@informatik.uni-wuerzburg.de

Published as invited talk at the 2005 Tyrrhenian International Workshop on Digital Communications

On the Stochastic Scalability of Information Sharing Platforms

Phuoc Tran-Gia and Andreas Binzenhöfer
Department of Distributed Systems
Institute of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{trangia,binzenhoefer}@informatik.uni-wuerzburg.de

Abstract

Recently, centrally controlled information distribution systems are rapidly emerging to decentralized structures. This tendency can be observed in software distribution applications using BitTorrent or information sharing platforms based on distributed hash table structures like Chord or Kademlia. To ensure that the emerging platforms will function properly with a growing number of users and services the issue of *scalability* turned into one of the hottest research topics.

Traditionally, the term scalability often restricts to the functional scalability, which describes the scalability in terms of the system size. In this regard the basic structure is stationary, i.e. it does not fluctuate frequently. However, when the stochastic behavior of system components, the network structure, and user applications has to be taken into account, the *stochastic scalability* has to be investigated in the context of performance evaluation. In this paper we discuss the stochastic scalability of information sharing platforms. We give a classification of current information sharing platforms and define the terms functional and stochastic scalability in detail. A distributed phone book based on a Chord ring will be discussed as an example to motivate other areas of application and to show the potential of the evaluation of stochastic scalability.

1 Information Mediation and Platforms

The scale of distributed systems has significantly changed with the growing size of the Internet in the last decade. Distributed applications have to serve thousands to millions of customers in parallel. Due to stochastic user behavior, the number of customers and their world wide locations, as well as the dynamic of current network architecture, such as overlay networks in file-sharing platforms, the underlying network structure and the usage can considerably change on different time scales. Together with the enormous growth of the size and the complexity of such systems, the need for information sharing platforms became immanently important. To guarantee the functionality of the system, investigations concerning the scalability of the application have to be carried out early during the conception and the dimensioning phases of the service deployment.

Concerning the architecture of information sharing platforms some current trends can be observed:

- *Transition to business cases*: Content distribution platforms and information sharing systems gain importance in the context of booming peer-to-peer (P2P) file sharing systems (e.g. Kaaza, eDonkey, ...). These applications are being transformed from a disruptive technology with rather gray-scale content (e.g. music downloads) to thoroughly designed business cases (e.g. distributed directory services, ...).

- *Information mediation*: In general, these systems can be categorized as information mediation platforms. The main task is similar to a telecommunication system: to efficiently mediate information from information providers to information consumers. Thus, the main trend is to move away from client-server based data centers or server farms to Internet oriented information storage and distribution services.

- *Distributed dynamic architecture*: Recently, centrally controlled information distribution systems are rapidly emerging to decentralized structures, which are usually implemented on a web-based platform. Examples are software distribution using BitTorrent or information sharing platforms based on distributed hash table structures like Chord [1] or Kademlia [2]. The structures of these systems are highly dynamic: during the system runtime, customers or network nodes can join or leave the system without notice. The system has to be designed to survive such so called "churns" with minimal service degradation.

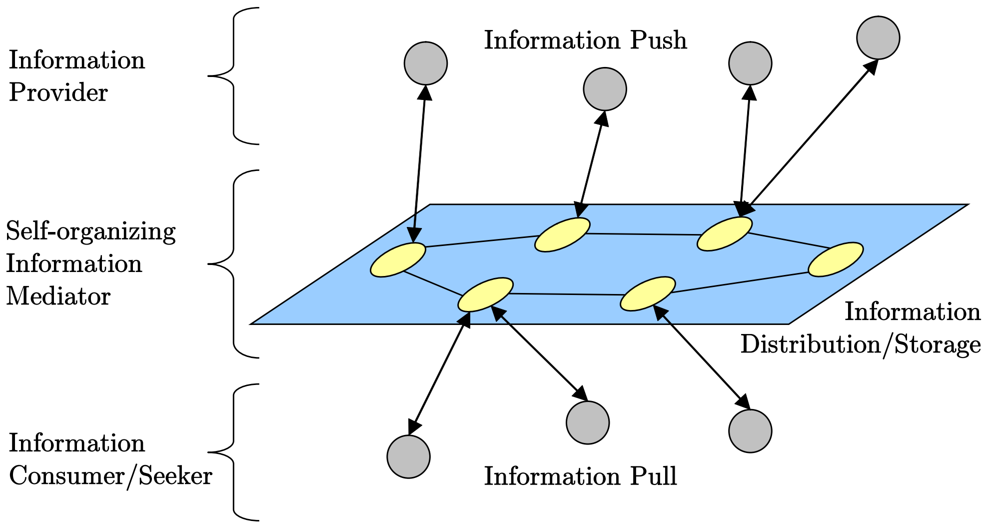

Figure 1: Overview of a self-organizing information mediator

As illustrated in Figure 1 most of recent distributed information sharing platforms [3] are characterized by three main components: The information provider, the self-organizing mediator and the information consumer or seeker:

- *Information providing site*: Information providers mediate data and resources using the information distribution plane. The information can be the varying profile of users in a distributed phone book embedded in a Voice-over-IP (VoIP) application. The profile contains the nickname of the user, his current IP-address, charging information, etc.
- *Information mediation plane*: The self-organizing information mediator itself can be used to store the information in a reliable and consistent way to distribute the data to the information consumer at a later point in time. Thus, the mediator has two basic (control) functions, the mediation of resources and the coordination of the resource access and exchange. It must control, schedule and conduct the exchange of resources in a scalable and efficient way. The information mediation plane is often referred to as "overlay network", like in P2P systems. In our VoIP example above, the information mediation plane can be built by a number of mediation nodes, which are connected using a Kademlia structure or a Chord ring. Due to the use of hash functions, the location of a profile, which is a search object for the information consumer, is well defined. Mediation nodes can be embedded in the system and are thus identical to customers.
- *Information consuming site*: The information consumer must be able to search and locate a resource at any given time. In a running environment a participant can be both information provider and information consumer at the same time. In the VoIP example a calling subscriber searching for the current location of a nickname is an information consumer. Using the search algorithm he can find out in which mediation node the information is stored.

The advantage of the described architecture is its bandwidth efficiency in the distribution of information. It autonomically enables fast access, resilience and scalability. However, so far there are no guarantees for security or data consistency and completeness.

In the traditional client-server architecture the server has the role of the information mediator. Current information sharing platforms, however, tend to rely on P2P overlay networks or mediation planes. The P2P paradigm reflects a highly distributed and adaptive application architecture. P2P systems solve two basic functions resource mediation, i.e. search for and location of resources, as well as resource exchange. The underlying P2P algorithms are highly efficient, scalable, and robust.

As illustrated in Figure 2, a P2P network builds a virtual overlay topology on top of an already existing IP network, like e.g. the Internet itself. These P2P overlays are increasingly used as a self-organizing and scalable information mediator. The first wave of P2P systems (∼1999), Napster being the most popular representative, relied on direct peer-to-peer communication and central index servers. The second wave, including P2P applications like KaZaA (2000∼2002) made use of supernodes and introduced unique file IDs using a hash function. The current wave (2003∼), enables fast and scalable resource discovery using distributed hash tables. The corresponding P2P protocols are suitable to serve as the information mediator in an information sharing platform.

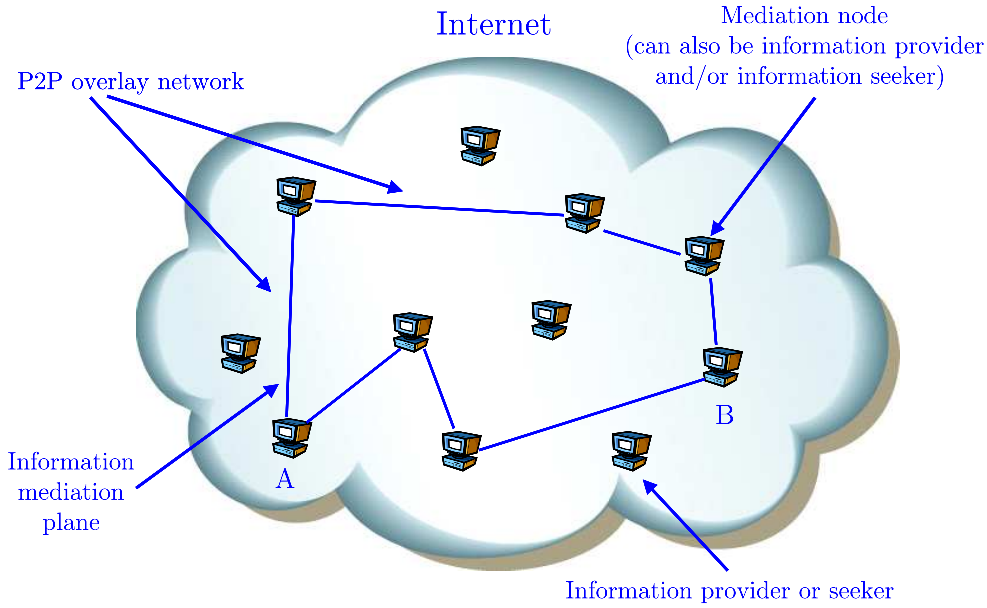

Figure 2: A P2P overlay network used as information mediator

However, due to their highly distributed application architecture those P2P systems are too complex for large scale emulation. Even simulations on packet level proved to be rather intractable. To better understand the dynamics of such systems one has to approach the problem on different levels of detail. While a detailed simulation can give valuable insights about the fundamental functionalities [4], a mathematical analysis of the main aspects of the problem often helps to investigate the scalability of the system itself [5, 6]. Due to the numerous stochastic processes involved in such highly distributed multi-user applications it is crucial to study not only the functional but also the stochastic scalability of the systems. In the following sections we define and further motivate the need for the evaluation of the stochastic scalability. Our goal is to better understand the dynamics of large scale information sharing platforms such as P2P systems. If we want to build reliable large scale information sharing platforms based on P2P mechanisms we need to master the complexity of such systems. Investigating the stochastic scalability, we will be able to get those systems under control and achieve carrier grade availability systems in a resource-efficient but also simple manner.

In this paper we refer to properties of the scalability of a system as stochastic scalability. One of this properties could be, to stay in our VoIP example, the quantile value, i.e. the time bound of search delays when we expect 99% (or even 99,99%) of searches to remain below this limit. Measures considering stochastic scalability can be used, e.g. to define and enforce Service Level Agreements (SLA) in communication systems and applications.

Traditionally, the term scalability often restricts to the functional scalability, which describes the scalability in terms of the system size, where the basic underlying application scenario and network structure are rather fixed or change only in a long-term time scale. More precisely, the question was: if a service or a solution, i.e. with a network carrying a target application, works properly for ten customers, will they also function accordingly for one thousand or for one million customers, following the potential growth of the market?

If the mid-term and shot-term stochastic behavior of system components and user applications has to be taken into account, the stochastic scalability has to be investigated in the context of classical performance evaluation. One possible question is: If a platform works properly (SLA is complied) in an environment with a network latency coefficient of variation $c_T = 0.5$, will it also support the same number of customers in a higher variation network with $c_T = 2$? This is crucial to ensure the network resilience in overload cases.

The remainder of this paper is structured as follows. In Section 2 we summarize the evolution of existing P2P-based information sharing platforms. Definitions of functional and stochastic scalability can be found in Section 3. A P2P-based VoIP solution is given as an example for the evaluation of the stochastic scalability of P2P-based information sharing platforms in Section 4. Section 5 finally summarizes and concludes the paper.

2 Evolution of P2P-based Information Sharing Platforms

2.1 The Traditional Client-Server Model

The client-server paradigm was the most prevalent model for information mediation in classical networks. It can be described as a service-oriented request-response protocol. A central server host runs a server process and provides access to a specific service such as web content or a centralized index. The client runs the corresponding client process and accesses the service offered by the central server host. Well known examples include but are not limited to protocols such as http or ftp. In the traditional client-server architecture the server had the role of the information mediator. It has become one of the central ideas of computer networks but severely suffers from two major drawbacks that come along with a centralized mediator:

- It represents a single point of failure. Once the central service provider, the server, fails, the offered service will be disrupted and no longer available to the customers, i.e. to the clients. The same problem could, e.g., be caused by a distributed denial of service attack which is targeted at a specific service. That is, the functionality of an entire business solution depends on the functionality of a single central unit.
- It hardly scales. The number of hosts that can be served at the same time is mainly restricted by two important properties of the central server: Its processing power and its available bandwidth. The latter is especially crucial in connection with the distribution of large files, such as software updates or multimedia content. The processing power of the central server might e.g. be the limiting factor when offering web services.

A new wave of highly distributed information mediation platforms emerged to cope with these problems and to provide reliable and scalable access to electronic information stored in a computer network. The distributed architecture of such platforms enables the offered service to be still available even if parts of the system crash or fail. The peer-to-peer paradigm plays an important role in this context and will be discussed in the next section.

2.2 P2P-based Information Sharing Platforms

A peer-to-peer (P2P) network can be described as a group of entities denoted as peers, with a common interest, that build a self-organizing overlay network on top of a mixture of already existing networks. That is, P2P is about the networked cooperation among equals. The main task is the discovery and sharing of pooled and exchangeable resources. An ever increasing number of companies discover the advantages of decentralized P2P networks. Skype [7], a P2P-based telephone directory, e.g., already attracts millions of users every day. P2P algorithms are also used to overcome the problems of distributed network management [8]. Due to the highly distributed application architecture companies using P2P mechanisms are no longer dependent on a single central unit nor do they have to invest in server farms to guarantee the scalability of their systems. Together with those new P2P systems, however, new challenges arise as well. In a business environment, they have to be able to guarantee efficient and, most important, scalable business solutions.

There are different P2P approaches trying to create information sharing platforms supporting tens and up to millions of entities to provide a highly scalable mediation platform. Those systems are designed to be highly dynamic, robust and resilient. In the ideal case any peer can be removed without resulting in any loss of service. P2P

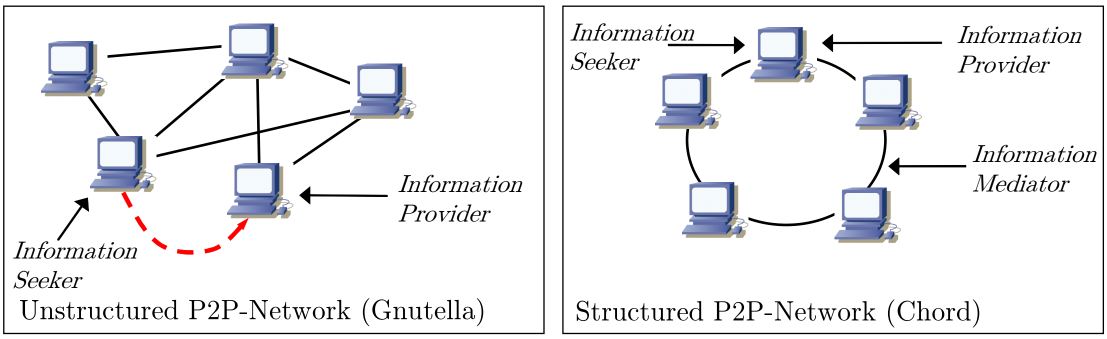

Figure 3: Unstructured and structured P2P overlay topologies

mechanisms can roughly be divided into two main classes, those that build structured and those that build unstructured overlay topologies. Figure 3 gives a visual comparison between the two classes. Unstructured P2P networks, like e.g. Gnutella [9], build an arbitrary overlay topology. Peers are randomly connected to each other resulting in a fully decentralized use of the overlay network paradigm. There are no dedicated peers that store specific information. That is, resources are located at arbitrary peers. To cope with this indetermination of the desired information, searches in unstructured P2P networks are performed using an expanding ring principle. A peer searching for information simply floods the query to all its neighbors in the overlay network, who in turn forward the query to all of their overlay neighbors until the desired information is eventually found. To keep the involved overhead traffic within reasonable limits a Time To Live (TTL) counter is associated with the query. The TTL is decreased on every hop, while query packets with a TTL of zero are simply discarded. It is easy to see, that such unstructured overlay topologies suffer from two main drawbacks. First, there is no guarantee that a search returns a positive result even if the searched information is stored on a large number of peers. Second and even more important, unstructured P2P mechanisms do not scale to a large number of information consumers.

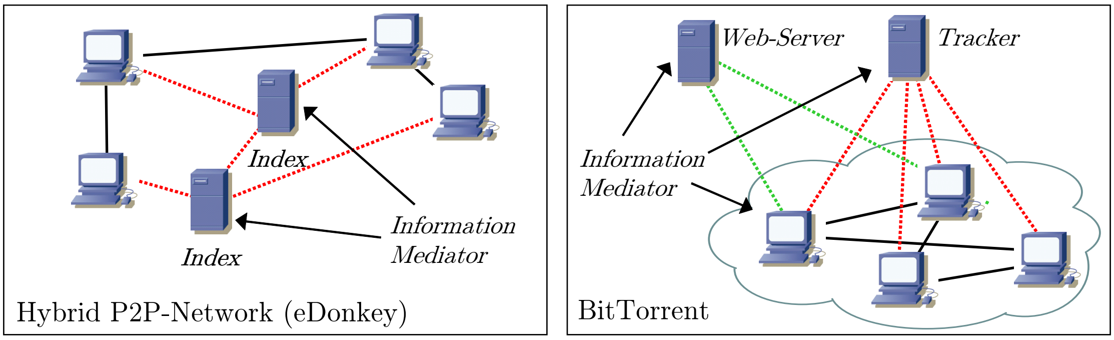

Figure 4: Hybrid P2P overlay topologies

Structured overlay topologies on the other hand build a network structure, which specifies communication relationship among the participating peers. The resource mediation can then take advantage of the structure in the overlay. Local information of a peer is sufficient to globally search for information and find resources in an efficient way. In difference to unstructured networks the search and routing processes are scalable, deterministic and always return positive results if the resource is available in the overlay. The predefined overlay structure, which in a generic way specifies the communication relationship among the peers, is usually realized using a distributed hash table (DHT). The best known DHT algorithm is Chord [1] which arranges the participating peers on a ring topology. The position on this ring is chosen according to the hash value of a unique attribute of each peer. The basic idea is that each peers has a good knowledge about its overlay neighbors, i.e. its predecessors and successors on the Chord ring, while only maintaining a few connections to more distant peers. This way the mediation of the information stored in the distributed network can be done using only $O(\log_2(n))$ messages to other peers, where n is the current size of the overlay network.

Besides the (un)structured topologies there are different hybrid overlay mechanisms, which partly use a structured control or rely on distributed index servers. The eDonkey overlay [10] is a classical example of a hybrid P2P network shown in the left part of Figure 4. While the information exchange is realized using direct P2P communication, the signaling and resource location relies on central index servers, which reflect the functionality of a classic server farm. There are also some more exotic overlay topologies serving a special purpose. BitTorrent, shown in the right part of Figure 4, e.g., is used for the rapid distribution of one single file. The underlying mechanism slices the information into small parts and uses multiple source download to mediate the information as fast as possible. The meta-information has to be stored on a central unit like a web server and a centralized tracker takes control of the coordination of the download process. The mediation of the file itself is done by distributed information transfer.

The size of the system and the behavior of the customers significantly affect the performance and the functionality of all above mentioned topologies. While the number of peers in the overlay topology interferes with the logic of the system, the stochastic and dynamic behavior of the user pushes the system to its limits and might even cause it to fail entirely. These procedures can be associated with stochastic and functional scalability which will be discussed in the next section.

3 Functional and Stochastic Scalability

Today, scalability is the most important performance measure a carrier grade system has to withstand. It indicates whether a system is going to work on a large scale or not. In general, the question scalability asks, is: If a solution works for 10 customers, does it also work for hundreds, thousands, or even millions of customers? So far, scalability mainly referred to the mere size of a system. Most studies were intended to determine if a system at hand does work for growing customer clusters. We summarize this kind of analysis under the term functional scalability. It tells us whether the fundamental logic of a solution is scalable.

The mere size of a system, however, is not the only factor in terms of scalability a running application has to cope with. There are more and more system parameters having a stochastic character. Consider, e.g., the stochastic behavior of customers. There are numerous different random variables describing values like the inter-arrival time, the mean on-line time, and the query rate of customers of large scale systems. In P2P networks this stochastic behavior is defined as the autonomy of the participating peers, i.e., the peers may join or leave the system arbitrarily. This leads to the requirement to evaluate P2P algorithms with respect to the stochastic on-line behavior, which is summarized under the term "churn" [11]. This unpredictable stochastic behavior of the end user results in a highly dynamic evolution of the P2P network and thus has a significant impact on the functionality of the system [12]. The customer, however, is not the only variable introducing probabilistic properties into the system. A running system also faces stochastic network loads, probabilistic variations in traffic volumes and random transmission delays, to name just a few. Thus, in order to provide stochastic scalability, P2P networks with resilience requirements have to be able to survive in case of stochastic breakdowns. Stochastic scalability can be analyzed combining methods and techniques of both probability theory and performance analysis.

Figure 5 visualizes the difference between functional and stochastic scalability. The functional scalability verifies whether the interworking logic is extendable to larger

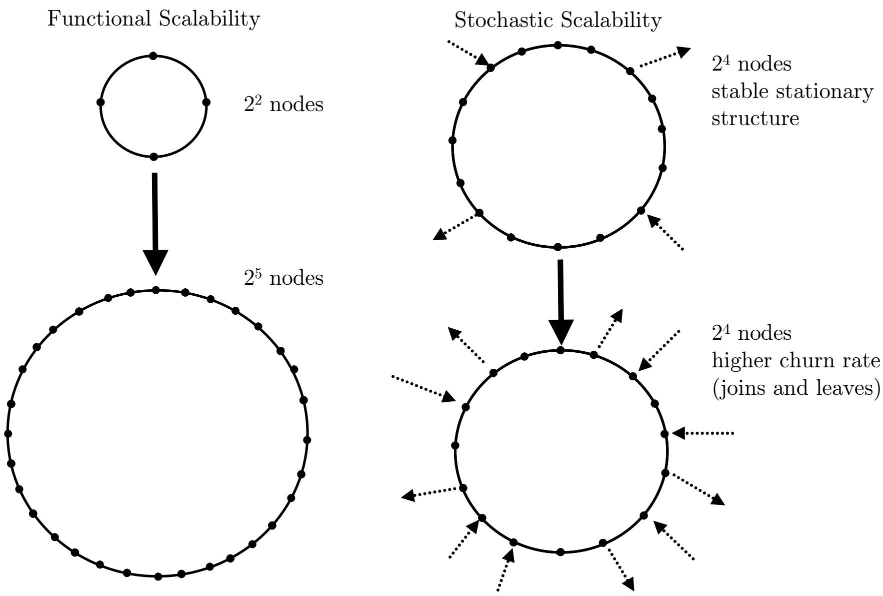

Figure 5: On the definition of stochastic scalability

crowds of customers. It mathematically analyzes whether the functionality of a system, like the search delay in the indicated Chord ring, also works for a large number of customers. Stochastic scalability on the other hand tries to verify whether a system can sustain the stochastic behavior of its components. It investigates whether a system can cope with the non-deterministic arrival, departure and query times of the participating customers. In respect of our Chord ring example stochastic scalability comprises the question whether a system, which can sustain minor churn rates, also works under extreme high churn rates? That is, we want to know how long the average customer has to stay on-line in order to guarantee the functionality of the running system.

In the end a successful system most be scalable in both a functional and a stochastic way. Without functional scalability a system will collapse under its own size, without stochastic scalability a system will collapse under the random variations of its components.

4 An Example of Stochastic Scalability

Due to the increased bandwidth of the end user there is a growing demand for the mediation of information especially in multimedia applications. In this context, more and more companies are using P2P mechanisms to realize their business solutions. Such P2P systems are, e.g., used for content distribution, as index servers or even for distributed network monitoring. At the moment the most predominant structured P2P architecture in the research community is the ring based Chord algorithm. While its main functionality is to store and retrieve key-value pairs, it can be used for a broad variety of applications. In this section we will have a closer look at the stochastic scalability of

Chord when used as a distributed phone book for voice-over-IP (VoIP) telephony. After a short description of IP telephony in general, we will explain how to use a P2P network in this context, define the problem areas, and show how to approach a performance analysis of the stochastic scalability of such a system.

4.1 P2P IP Telephony

Traditionally, telephony was the domain of large telecommunication carriers. Telephone calls were made from on telephone set to another telephone set using hardware and switching centers provided by the telecommunication carriers. With the introduction of broadband Internet, if not before, new possibilities to make voice calls arose. With applications like Microsoft Netmeeting, VoIP calls from computer to computer using headsets became possible. Not until recently, the first companies discovered the advantages of the packet switched Internet over the old public switched telephone network (PSTN). Companies like Net2Phone offer ways to make calls from the Internet to regular telephone sets. Other companies like Sipgate even introduce the possibility to place calls from a regular telephone line to an Internet user.

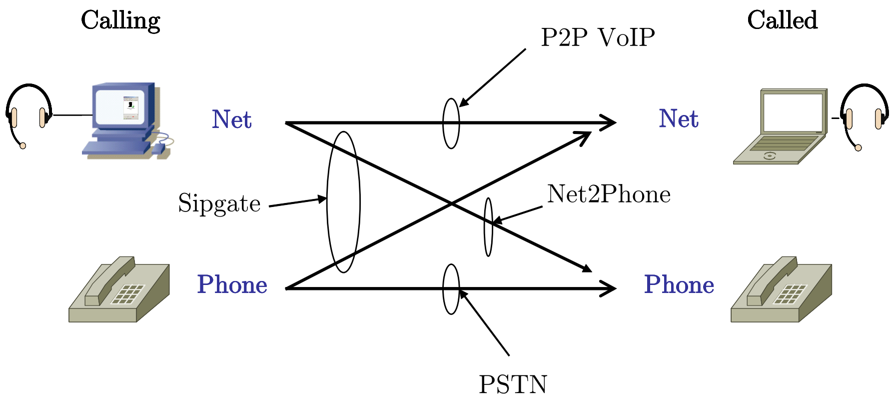

Figure 6: Different approaches to transmit voice from the caller to the callee

The different ways of making telephone calls between PSTN and Internet users are summarized in Figure 6. Even so direct calls between Internet users are realized transmitting datagrams over IP networks, they still rely on a central unit, which is in charge of phone book lookups. In the meanwhile, however, highly distributed P2P-based VoIP solutions emerge, which do no longer rely on any central unit. The phone book is now realized in a distributed way, using a P2P-based information sharing platform. The most prominent example of a P2P-based VoIP telephony application is Skype. It offers free calls between Internet users, cheap calls from the Internet into the PSTN using the SkypeOut service and even calls from the PSTN to the Internet using the SkypeIn service. The advantages of a P2P-based VoIP solution are obvious. They are highly scalable, do not need any concentrated processing power, nor do they suffer from a single point of failure. In addition they are very inexpensive for the end user.

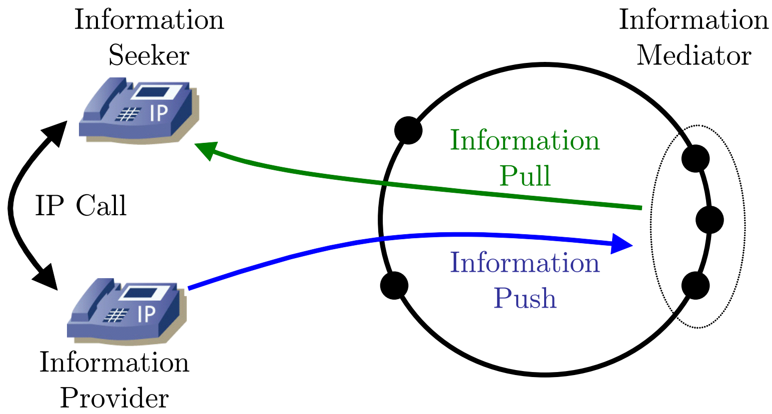

Figure 7: Call setup using a Chord based information mediator

From a technical point of view, the main difference between a central and a P2P-based VoIP solution can be found in the call setup. In a P2P-based solution, the P2P overlay network is used to realize a distributed phone book, which in this case represents the information mediator. If a VoIP customer wants to publish his personal phone book entry, he becomes an information provider and stores his contact information in the mediation plane. When another VoIP customer wants to call this client at a later point in time, he takes the role of an information seeker and retrieves the corresponding contact information from the information mediator. So far, however, P2P-based IP telephony solutions come without guarantees for data consistency and security in the information plane. Furthermore, they involve signaling and information exchange overhead to maintain a consistent view of the stored resources. If a telecommunication carrier intends to build a large scale, P2P-based application, it must meet higher demands than a best effort service. The reachability of the customers has to be guaranteed. In this case, the functional scalability of the system alone does not suffice, since it only guarantees the scalability in terms of system size. The telecommunication operator, however, wants to be able to guarantee a certain quality of service. It should, e.g., be guaranteed that 99.99 percent of all call setups can be completed within a certain time limit. This, however, highly depends on a number of stochastic processes. The network transmission delay, e.g., can be regarded as a random variable. To be able to make any quality of service statements one needs to know the entire distribution function of the call setup delay. In the following example we show how to prove the stochastic scalability of the search delay in a Chord-based information sharing platform, calculating the quantiles of the search delay in such a system.

4.2 Performance Analysis of a VoIP Platform

In this section we show how to analyze the stochastic scalability of a Chord-based information sharing platform. The results can be used to realize a phone book for a P2P-based VoIP application. In particular, we analyze the time needed to complete a search in a Chord-based P2P system. Since the physical path delay strongly influences the performance of searches in such P2P systems, the stochastic impact of network delay variation is taken into consideration. The following random variables describe some of the stochastic processes, which are involved in a search for resources:

T_N: describes the delay of a query packet, which is transferred from one peer to another peer

T_A: represents the time needed to transmit the answer from the peer (having the answer) back to the originator

T: describes the total search duration

X: indicates how many times a query has to be forwarded until it reaches the peer having the answer. X will be denoted as the peer distance

H: number of overlay hops needed to complete a search, i.e. the number of forwards of the query plus one hop for the transmission of the answer

n: size of the Chord-based P2P system

The search process is visualized in Figure 8. The peers connected by the blue lines build the P2P overlay network according to the Chord algorithm. In the example peer A is searching for information stored on peer B. Peer A sends a query, which will recursively be forwarded until it finally reaches peer B. Each of the X overlay hops can be described by the random variable T_N.

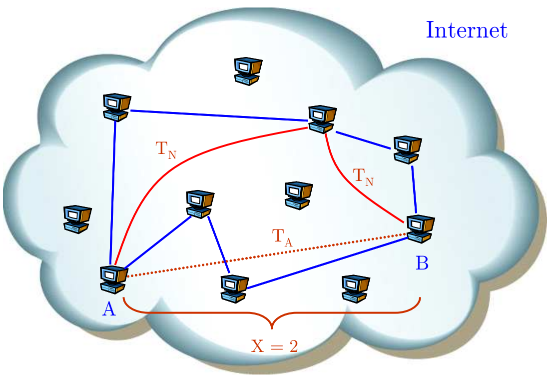

Figure 8: Model of the search in a P2P network

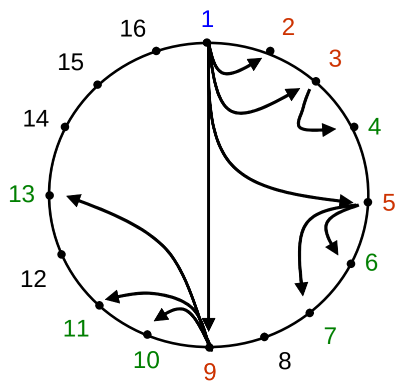

Figure 9: The model applied to Chord

Figure 9 shows the extended model of the search applied to the Chord algorithm. In this example the peer with $id_p = 1$ issues some queries for other peers. According to the Chord algorithm peers 2, 3, 5, and 9 can be reached using only $X = 1$ overlay hop. Furthermore, it takes $X = 2$ overlay hops to reach peers 4, 6, 7, 10, 11, 13. Extending this model, we are able to calculate the number of hops needed to reach a peer, that

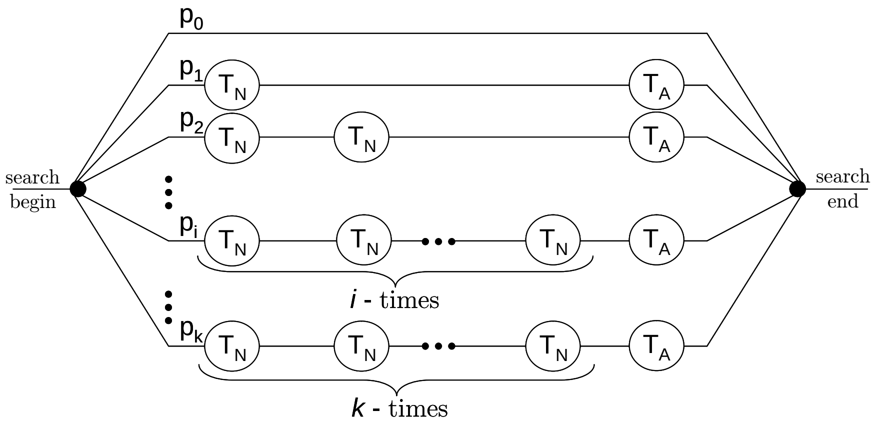

Figure 10: Phase diagram for the information pull in a call setup

answers a specific query. We are thus able to derive the probability $p_i = P(X = i)$ that the searched peer is exactly i hops away from the searching peer. The detailed analysis can be found in [13].

Knowing the peer distance distribution X, we derive the length in hops of the path a particular search-query takes through the overlay network. Together with the probability p_i, that a search takes the corresponding path through the overlay, we can then compute the entire distribution of the search delay as a function of the stochastic network delay characteristics.

The phase diagram of the search delay is depicted in Figure 10. A particular path i is chosen with probability p_i where phase i consists of i network transmissions T_N to forward the query to the closest known finger and one network transmission T_A to send the answer back to the searching peer as illustrated in Figure 8. By means of the phase diagram, the generating function and the Laplace-Transform respectively can be derived to cope with the case of discrete-time or continuous-time network transfer delay:

$$X(z) = p_0 + \sum_{i=1}^{k} p_i \cdot X_A(z) \cdot X_N^i(z)$$

and the Laplace-Transform

$$\Phi(s) = p_0 + \sum_{i=1}^{k} p_i \cdot \Phi_A(s) \cdot \Phi_N^i(s).$$

The mean and the coefficient of variation of the search delay are such:

$$E[T] = \sum_{i=1}^{k} p_i \cdot (E[T_A] + i \cdot (E[T_N]))$$
$$E[T^2] = \sum_{i=1}^{k} p_i \cdot (VAR[T_A] + i \cdot VAR[T_N] + (E[T_A] + i \cdot E[T_N])^2)$$

and

$$c_T^2 = \frac{E[T^2] - E[T]^2}{E[T]^2}.$$

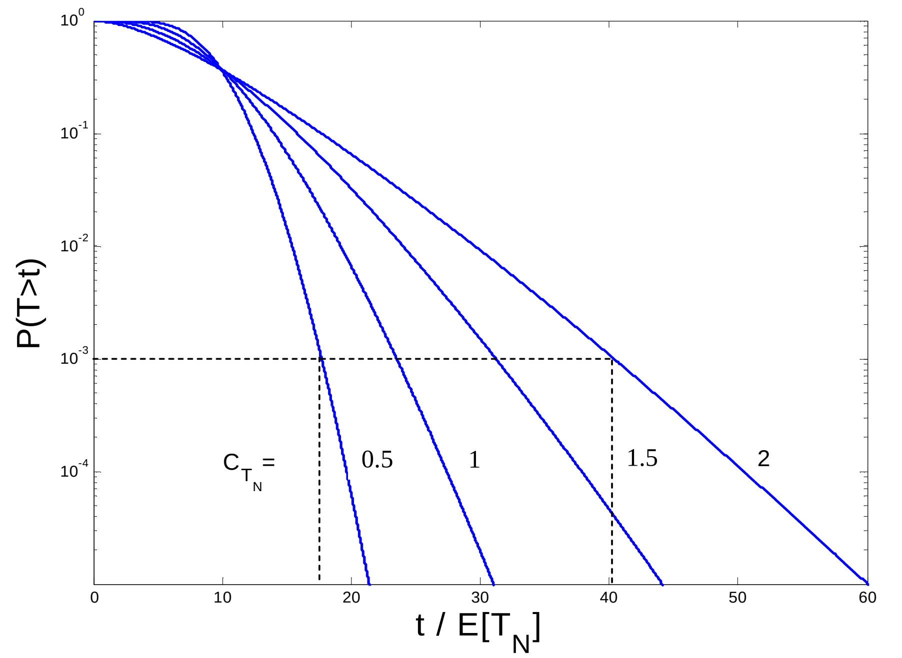

Figure 11: Distribution of the search delay for different values of c_{T_N}

Other than the functional scalability the telecommunication carrier is now especially interested in the stochastic scalability of the system. In this context, the stochastic component with the most significant impact on the search delay is the variation of the one-hop delay T_N. This variation can be expressed by the corresponding coefficient of variation c_{T_N}. To analyze the stochastic scalability, we set the network size and the mean of the one-hop delay to a fixed value and concentrate on the coefficient of variation of the one-hop delay c_{T_N} as a parameter. That is, instead of the size of the system the

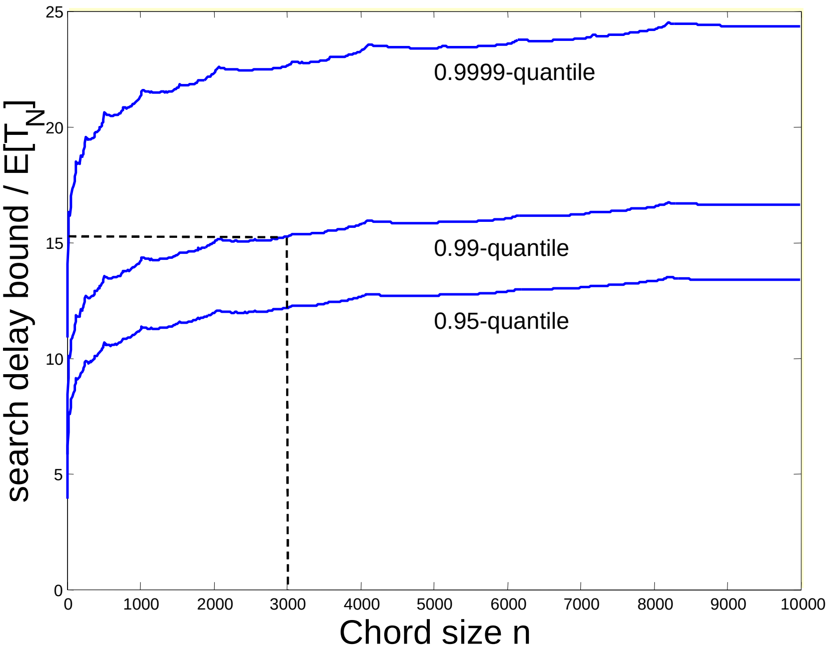

Figure 12: Different quantiles for the search delay in a distributed phone book

stochastic influence of the search delay increases. In Figure 11 we analyze the impact the stochastic variation of the network delay has on the search delay. We depict the entire inverse search delay distribution for different values of c_{T_N}.

Note, that the search delay increases for larger values of c_{T_N}. The values $P(T > t)$ on the y-axis indicate how many percent of all searches will take longer than the corresponding time on the x-axis. Accordingly, $1 - P(T > t)$ percent of all lookups will take less time than the corresponding value on the x-axis. As indicated by the dotted lines, the value 10^{-3} on the y-axis, e.g., indicates that 99.9 percent of all phone book searches take less than roughly 18 overlay hops in the case of $c_{T_N} = 0.5$ and about 40 hops in the case of $c_{T_N} = 2$. Note, that since the results are independent of the mean value of the one-hop delay the values on the x-axis are normalized by $E[T_N]$. That is, in the case of $E[T_N] = 50$ms, e.g., 18 hops correspond to 900ms. In this scenario, it would therefore take 99.9 percent of all costumers less than 900ms to find their VoIP communication partner given that $c_{T_N} = 0.5$.

In a real business case, however, the operator of the system needs to assure both the functional and the stochastic scalability at the same time. In particular, he wants to know a search delay bound, which will be met by say 99.99 percent of all queries issued in the system independent of the current size of the system. Due to the mathematic analysis of the underlying stochastic processes it is possible to prove the functional and the stochastic scalability of the search delay. Figure 12 depicts the quantiles of the search delay T again normalized by $E[T_N]$. Next to the mean delay, which shows the functional scalability, different quantiles for the search delay are taken as a parameter.

The curve with the 99%-quantile, e.g., indicates that 99 percent of search durations lie below that curve. For a peer population of, e.g., $n = 3000$ in 99 percent of all cases the search delay is less then roughly 15 times the average network latency. It can be seen that the curves indicate stochastic bounds of the search delay. This can be used for dimensioning purposes, e.g. to know the quality of service in a search process with real-time constraints like looking at a phone directory, taking into account the patience of the users. Compared to the mean of the search delay the quantiles of the search delay are on a significantly higher level. Still the search delay scales in an analogous manner for the search delay quantiles. The above example shows the importance of stochastic scalability in the network planning process of a telecommunication carrier. Needless to say, that a comprehensive study of the stochastic scalability of a real system is more complex than the above example. The stochastic churn behavior of the participating peers, e.g., also has a great influence on the functionality of a running system. The more frequently customers enter and leave the system, the harder it is to maintain a stable overlay structure and the more timeouts will occur during a search process.

5 Conclusion

In this paper we gave a brief introduction to existing information sharing platforms. In particular, we provided a simple classification as well as a short description of P2P mechanisms. Current P2P algorithms are thought to be scalable and robust enough to serve as mediation platforms for highly distributed applications like VoIP solutions without any central unit. However, we showed that in this context the term scalability requires a more exact definition.

In addition to functional scalability, we introduced the stochastic scalability for the performance evaluation of large scale telecommunication systems. It regards the probabilistic behavior of system influence factors, like the average on-line time of a user or the variation of the network transmission delay. Using an example, we further motivated the need to consider stochastic scalability in the performance evaluation of current information sharing platforms. Stochastic influences play a decisive role in today's telecommunication systems and will also be one of the crucial factors in solutions of the next generation.

Acknowledgements

The authors would like to thank Robert Henjes, Tobias Hoßfeld, Dirk Staehle and Kurt Tutschku for the help and discussions during the course of this work.

References

[1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in *ACM SIGCOMM 2001*, (San Diego, CA), August 2001.

[2] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *IPTPS 2002*, (MIT Faculty Club, Cambridge, MA, USA), March 2002.

[3] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. marie Kermarrec, "The many faces of publish/subscribe," in *ACM Computing Surveys*, 2003.

[4] T. Hoßfeld, K. Tutschku, F.-U. Andersen, H. de Meer, and J. Oberender, "Simulative performance evaluation of a mobile peer-to-peer file-sharing system," in *Next Generation Internet Networks NGI2005*, (Rome, Italy), April 2005.

[5] A. Binzenhöfer, D. Staehle, and R. Henjes, "On the stability of chord-based p2p systems," Tech. Rep. 347, University of Würzburg, 11 2004.

[6] K. Tutschku and P. Tran-Gia, "Traffic characteristics and performance evaluation of peer-to-peer systems," in *Peer-to-Peer-Systems and Applications* (K. W. Ralf Steinmetz, ed.), Springer, 2005.

[7] Skype. URL: http://www.skype.com.

[8] A. Binzenhöfer, K. Tutschku, and B. auf dem Graben, "DNA – A P2P-based Framework for Distributed Network Management," in *Peer-to-Peer-Systeme und -Anwendungen*, (GI/ITG Work-In-Progress Workshop in Cooperation with KiVS 2005, Kaiserslautern), March 2005.

[9] Gnutella website, "http://www.gnutelliums.com."

[10] eDonkey. URL: http://www.edonkey2000.com/.

[11] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *2004 USENIX Annual Technical Conference*, (Boston, MA), June 2004.

[12] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Observations on the dynamic evolution of peer-to-peer networks," in *International Workshop on Peer-to-Peer Systems (IPTPS '02)*, (Cambridge, MA), March 2002.

[13] A. Binzenhöfer and P. Tran-Gia, "Delay Analysis of a Chord-based Peer-to-Peer File-Sharing System," in *ATNAC 2004*, (Sydney, Australia), December 2004.