 The definitive version of this paper has been published in GLOBECOM 2005, 2005, 10.1109\/glocom.2005.1577765.

On the Stability of Chord-based P2P Systems

Andreas Binzenhöfer, Dirk Staehle and Robert Henjes
Department of Distributed Systems, Institute of Computer Science
University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{binzenhoefer, staehle, henjes}@informatik.uni-wuerzburg.de

***Abstract*— The current generation of P2P networks is intended to provide cost-effective alternatives to the traditional client-server architecture. The main goal is to store and retrieve data in a decentralized manner. The challenge in doing so consists in creating a stable overlay network that allows for fast and efficient searches. In this paper we consider the Chord P2P algorithm in this context and analyze its stability and efficiency with stochastic methods. We present realistic probabilities for a disconnection and investigate the corresponding scalability.**

I. Introduction

Current peer-to-peer (P2P) networks like gnutella, eMule and KaZaA are primarily used to share movies, music and the like [1]. Meanwhile, however, the first business models based on P2P architectures have emerged. Companies start to discover the advantages of decentralized P2P networks. They 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. Thanks to a new generation of structured P2P systems, based on Distributed Hash Tables (DHTs), distributed storage systems or distributed indexes become possible. Together with those new systems, however, new challenges arise as well. The main challenge of a DHT is to guarantee a consistent global view of the stored data. However, the stability of P2P overlay networks is strongly affected by the dynamic behavior of the end user. When many peers leave simultaneously, the overlay may be split into several disjoint networks or even collapse entirely. In case of such an inconsistent overlay state, successful searches can no longer be guaranteed and it might even not be possible to reestablish a stable overlay network again. An analysis of the evolution of such systems can be found in [2] and [3]. In this paper we concentrate on Chord [4], a DHT based P2P algorithm, and analyze the way it preserves reachability and stability of its overlay network. The stability of a Chord-based P2P system depends on the number of overlay connections a peer maintains. In contrast to previous studies [4] we show that the probability to lose the overlay structure of a DHT is not negligible in all cases. In particular, we present an analytical expression that can be used to calculate the probability to lose the routing functionality of a DHT given a certain number of overlay connections. We are able to evaluate the consequences of maintaining too many or too few overlay connections in a running system. The analysis can also be used to compute the actually necessary number of overlay connections to guarantee a stable overlay network.

The remainder of this paper is structured as follows. In Section II we describe the basic ideas behind Chord that are relevant to our analysis. In particular, we summarize how Chord realizes robustness and overlay stability. In Section III the probability for a loss of the overlay stability in Chord rings is calculated and strengthened by more realistic failure probabilities in Section IV. The results of our analysis are presented in Section V and Section VI finally concludes this paper.

II. Chord basics

The main purpose of P2P networks is to store data in a decentralized overlay network. Other peers will then be able to retrieve this data using the corresponding search algorithm. The Chord algorithm solves this problem by arranging the participating peers in a ring structure. The position of a peer on this overlay ring is determined by an m-bit identifier using a hash function such as SHA-1 or MD5. Additionally, each document that is to be stored in the P2P network is assigned an m-bit identifier using the same hash function. Based on these identifiers the underlying P2P mechanism decides where to store the documents. That is, the P2P algorithm determines which peers are going to be responsible for which documents. Peers searching for particular documents will then use the same algorithm to retrieve the searched information from the P2P overlay network.

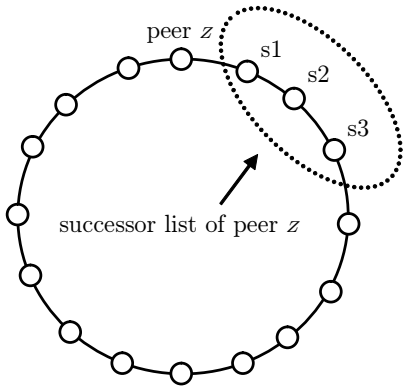

Fig. 1. A Chord ring of size $n = 16$, each peer maintains $r = 3$ successors.

To maintain the ring structure of the overlay, each peer stores pointers to the first r successors on the ring, i.e. the first r peers that follow the peer in a clockwise direction on the ring. Thus, if one of the peer's r successors goes offline, the peer will still know the next $r - 1$ peers on the

ring. If a peer, however, loses all its r successors, the ring will be disconnected. According to [4] the connectivity of the Chord ring can be obtained with high probability as long as $r = \Omega(\log_2(n))$, where n is the current size of the Chord ring[1].

Fig. 1 shows a snapshot of a Chord ring consisting of 16 overlay peers. Peer z maintains a successor list of size 3 consisting of peers s1, s2 and s3.

III. Validation of Chord's Stability

A P2P overlay network is connected if there exists a route from every peer to every other peer. Running the Chord algorithm, each peer maintains a list of $O(\log_2(n))$ successors to keep the overlay connected. A peer gets disconnected from the network if all of its successors fail. According to [4] a Chord overlay network stays connected with high probability, even if every peer fails with probability $\frac{1}{2}$. The proof relies on the fact that even though every individual peer fails with probability $\frac{1}{2}$, it is very unlikely that all $O(\log_2(n))$ successors of a peer fail at the same time. The conclusion that thus all peers stay connected with high probability, however, misses a subtle point. Although a local disconnection (one specific peer loses all its successors) might be very unlikely, one can not draw the conclusion that a global disconnection (at least one peer in the overlay loses all its successors) is very unlikely as well. To gain a better understanding of this subtle but important point, we introduce some definitions:

- p_{fail}: probability that a node fails
- $p_{ld}(r)$: probability that a specific node loses all its r successors and gets locally disconnected
- $p_{gd}(n, r, p_{fail})$: probability of a global disconnection, i.e. the probability that at least one peer gets locally disconnected in a network of size n, where each node knows r successors and each node fails with probability p_{fail}

The probability for a local disconnection can then easily be calculated as

$$p_{ld}(r) = p_{fail}^r \tag{1}$$

Obviously, the more successors a peer has, the less likely it gets locally disconnected. Since peers usually maintain a successor list of size $r = O(\log_2(n))$, a local disconnection is less likely in larger networks[2]. However, based on this observation alone, we can not conclude that the probability of a global disconnection is comparably small as well. The more nodes there are in the overlay network, the higher the probability that at least one of them gets locally disconnected. In other words, there is a trade-off between these two mechanisms. On the one hand, the larger the Chord ring becomes, the more successors are maintained by a peer, resulting in a smaller probability for a local disconnection. On the other hand, the larger the Chord ring becomes, the more peers run the risk of getting locally disconnected, resulting in a higher probability for a global disconnection.

To estimate the stability of a Chord ring, we need to calculate the probability $p_{gd}(n, r, p_{fail})$ of a global disconnection, i.e. the probability that at least once r or more contiguous peers fail on the Chord ring. As an approximation we neglect the ring structure of the overlay network and imagine the overlay peers arranged in an ascending row as shown in Fig. 2. We regard the probability $p_{rd}(x, r, p_{fail})$ that at least

Fig. 2. The n peers of a Chord ring arranged in an ascending row.

once r or more contiguous peers fail in such a row of x peers. Moreover, we can assume a random distribution of failures, since the hash function distributes peers equally in the identifier space and physical proximity therefore does not reflect overlay proximity. For the sake of simplicity, we use the short notation $p_{rd}(x)$ instead of $p_{rd}(x, r, p_{fail})$ where appropriate. Obviously, the probability that r or more peers fail in a row of less than r peers is zero, as indicated by the dotted peers in Fig. 2. If we consider the same probability in a row of exactly r peers, all peers have to fail accordingly. The corresponding equations are:

$$p_{rd}(x, r, p_{fail}) = 0 \qquad \text{if } x < r \tag{2}$$

$$p_{rd}(x, r, p_{fail}) = p_{ld}(r) = p_{fail}^r \qquad \text{if } x = r \tag{3}$$

In case of $x > r$ we obtain:

$$p_{rd}(x) = p_{rd}(x-1) + (1 - p_{rd}(x-r-1)) \cdot (1 - p_{fail}) \cdot p_{ld}(r) \tag{4}$$

The probability p_{rd} is defined recursively. To calculate $p_{rd}(x)$, we take the probability $p_{rd}(x-1)$ that there was at least one local disconnection in the first $x-1$ peers and add the probability that the first local disconnection occurs at peer x. The second term of this sum is best explained using Fig. 3. There are two requirements in order that the first local

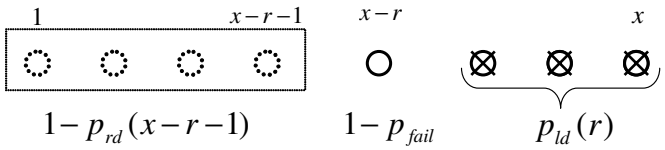

Fig. 3. Probability, that the first local disconnection occurs at peer x.

disconnection occurs exactly at peer x. First of all, there must not be a local disconnection in the first $x-r-1$ peers as indicated by the box in Fig. 3. Secondly, peer $x-r$ must not fail, while all of the last r peers have to fail to cause the disconnection at peer x.

According to Eq. (2) and Eq. (3), the first local disconnection can occur at peer r. Thus, there are still $r-1$ peers that could experience a local disconnection but are not

[1]Definition: $T(n) = \Omega(f(n))$ if and only if there are constants c_0 and n_0 such that $T(n) \geq c_0 f(n) \; \forall \; n \geq n_0$

[2]See [5] for a discussion of how to estimate the size n of the current overlay network

accounted for in our equation. To improve the accuracy of our approximation, we add $r-1$ peers at the end of the row as shown in Fig 4. Thus, there are n peers in a row of $n+r-1$

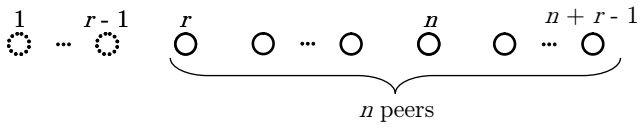

Fig. 4. To incorporate the first $r-1$ peers, they are added at the end of the row.

peers that can experience a local disconnection. The resulting approximation for the probability of a global disconnection in a Chord ring of size n is:

$$p_{gd}(n,r,p_{fail}) \approx p_{rd}(n+r-1,r,p_{fail}) \tag{5}$$

The reason for the approximation is that we neglect the ring structure of the overlay network. In fact the probability is slightly overestimated since the $r-1$ peers we added at the end of the row are obviously correlated with the first $r-1$ peers in the row. That is, there are some failure patterns in the last $r-1$ recursion steps that have already been taken into account before and are thus counted twice.

In Section V we validate Eq. (5) by simulation and present realistic probabilities of a global disconnection. Note that the formula is not limited to the special case of

$$r = \lceil \log_2(n) \rceil.$$

In fact, we are able to evaluate the consequences of using too large or too small values for r, i.e. of using more or less than $\log_2(n)$ successors.

IV. Realistic Failure Probabilities

In the previous section we were simply assuming values for p_{fail}, the probability that a node fails. In practice, however, there is not much sense in saying a node fails with a certain probability, without specifying a corresponding time frame. To guarantee overlay stability, a Chord peer refreshes its successor list every t_{stab} seconds by periodically calling a *stabilize()* procedure. This *stabilize()* function takes care that a peer's successor list is up to date by merging its list with the list of its closest successor. Thus, a peer gets locally disconnected if all of its known successors go offline between two *stabilize()* calls. Therefore, one should consider the probability that a peer fails within this periodic update interval instead of assuming some arbitrary values for p_{fail}.

On account of this, we regard the average online time of a peer E_{on} in seconds. Assuming that the online time is exponentially distributed with $\lambda_{on} = \frac{1}{E_{on}}$ it follows that

$$A(t) = 1 - e^{-\lambda_{on} t} \tag{6}$$

is the distribution function of the online time of a single peer. Due to the memoryless property of the exponential distribution the probability that a peer goes offline within t_{stab} seconds is[3]:

$$p_{fail} = A(t_{stab}) = P(A \leq t_{stab}). \tag{7}$$

We can then use this p_{fail} in Eq. (5) to calculate the probability of a global disconnection within t_{stab} seconds. The probability of a global disconnection increases with the number of *stabilize()* calls. The longer the Chord ring exists, the greater the probability of a global disconnection within its lifetime becomes. The probability $p_{it}(n,i)$ that a Chord ring of size n gets globally disconnected sometime within i *stabilize()* calls can be calculated as follows:

$$p_{it}(n,i) = 1 - (1 - p_{gd}(n,r,p_{fail}))^i. \tag{8}$$

In Section V we present a parameter study, covering reasonable values for t_{stab}, E_{on} and r, the current size of the successor list. We also show the impact of realistic failure probabilities on the probability of a global disconnection and analyze how this probability increases over time.

V. Numerical Results

In this section we concentrate on results regarding the problem of a disconnection. At first we have a closer look at the probability of a local disconnection. Figure 5 illustrates the probability of a local disconnection (cf. Eq. (1)) against the overlay size for three different failure probabilities of a peer. The number of successors is thereby set to $\lceil \log_2(n) \rceil$. As expected the probability of a local disconnection strongly decreases with the size of the overlay network. This is obviously due to the fact that a peer maintains more successors in larger networks and is thus less likely to be disconnected. Note that in a ring of size $n = 10^6$ and a failure probability of $p_{fail} = \frac{1}{2}$ we have a very low probability of a local disconnection of about 10^{-6}.

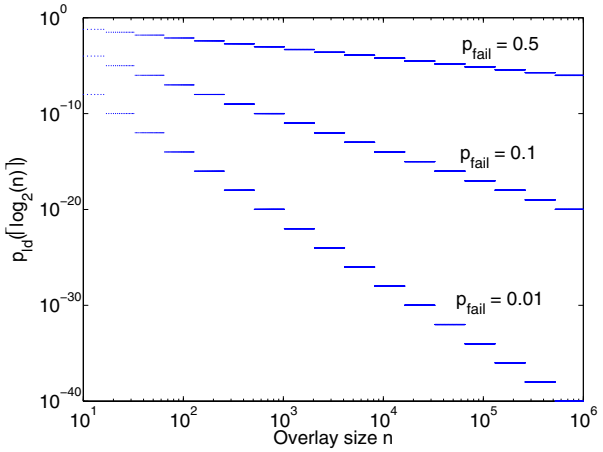

Fig. 5. Probability of a local disconnection for different values of p_{fail}

To show that based on these facts alone, we can not derive a very low probability for a global disconnection as well,

[3]The probability that a peer goes offline and online again within t_{stab} seconds is neglected in this context.

we calculate the probability of a global disconnection for $p_{fail} = \frac{1}{2}$. Fig. 6 shows this probability (cf. Eq. (5)) for networks of size $n = 2^k$, where each peer maintains a successor list of size $r = \log_2(n) = k$. The probability of a global disconnection does indeed decrease with the size of the overlay network. However, it does not approach zero but asymptotically reaches a probability of about 40 percent. So when every node fails with probability $p_{fail} = \frac{1}{2}$ and every peer maintains a successor list of size $r = \log_2(n)$ Chord does not stay connected with very high probability but gets disconnected with a probability of roughly 40 percent.

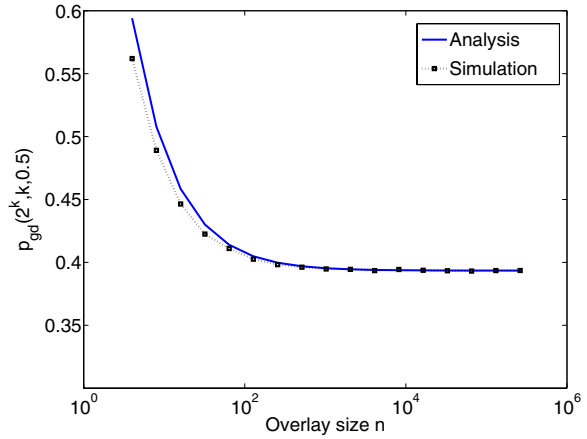

Fig. 6. Probability of a global disconnection in the special case of $p_{fail} = \frac{1}{2}$.

To confirm this result we simulated the probability of a global disconnection by generating snapshots of rings of a specific size and counting the percentage of those rings that did not get disconnected after 50 percent of all peers failed. The simulations were repeated until the confidence intervals became smaller than 0.001. For smaller values of n the results obtained by our analysis are slightly above the simulated values as the analysis does not take the ring structure into account. The error becomes negligible for overlay sizes above $n = 100$.

In practice, however, a failure probability of $p_{fail} = \frac{1}{2}$ is obviously too pessimistic. To obtain realistic values for p_{fail} we evaluate Eq. (7) for different average online times of a peer and different values of t_{stab}. Fig. 7 shows that even if the average peer only stays online for 10 minutes and successor lists are only refreshed every 60 seconds, the probability that a peer fails within this frame of time is still less than 10 percent.

In the following analysis we therefore concentrate on p_{fail}= 0.1, 0.05 and 0.01. Fig. 8 illustrates that a global disconnection is very unlikely for these values of p_{fail}. Even for a peer failure probability of 10 percent, a Chord ring of size 10^5 will be globally disconnected with a probability of less than 10^{-12}. The staircase shape of the curve arises from the fact that the plot is done for arbitrary n and corresponding successor lists of size $r = \lceil \log_2(n) \rceil$. So whenever the overlay size n crosses a power of two, each peer starts to maintain one additional successor in its successor list. Therefore, the probability of

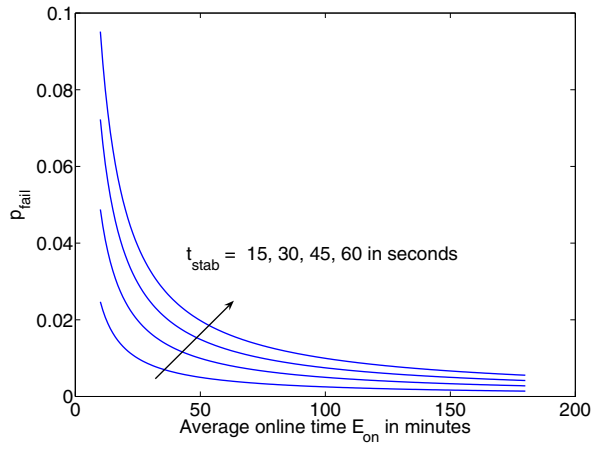

Fig. 7. Failure probabilities in dependency of the average online time of a peer

a disconnection abruptly decreases whenever a power of two is exceeded. It then slightly increases until the next power of two, since the probability of a local disconnection stays the same, but there are more peers that can get disconnected and cause a global disconnection.

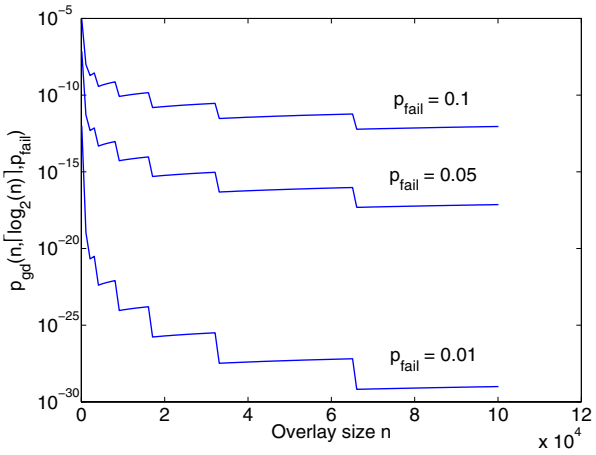

Fig. 8. Probability of a global disconnection maintaining a successor list of $\lceil \log_2(n) \rceil$.

So far, the results relied on a dynamic adaptation of the size of a peers successor list. In practice, however, it is more common to choose a fixed successor list size a priori. Fig. 9 illustrates the probability of a global disconnection for fixed successor list sizes of 3, 6 and 9. The failure probability of a peer is set to $p_{fail} = 0.01$. As we can see, the probability of a disconnection increases with the overlay size but scales very well to larger networks. Moreover, the order of magnitude of the probability of a global disconnection can be adjusted by choosing the corresponding successor list size. Obviously, less than $\lceil \log_2(n) \rceil$ neighbors are sufficient to guarantee a stable Chord ring when we assume a realistic failure probability of $p_{fail} = 0.01$.

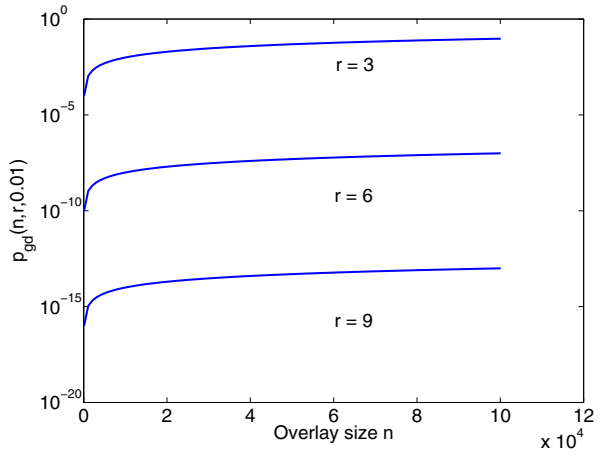

Fig. 9. Impact of a fixed number of successors on the global disconnection.

To illustrate the effects of extremely high failure probabilities we plot the probability of a global disconnection against the number of successors r. In Fig. 10 we show the results for a peer failure probability $p_{fail} = \frac{1}{2}$ and three different ring sizes $n = 2^5$, 2^{10} and 2^{15}. The vertical black dotted lines represent the suggested successor list size $\lceil \log_2(n) \rceil$. Again, the suggested number of successors results in a disconnection probability of about 40 percent. To guarantee a global disconnection probability close to zero in this example, a peer has to maintain a successor list of size $\lceil \log_2(n) \rceil + 7$ or more.

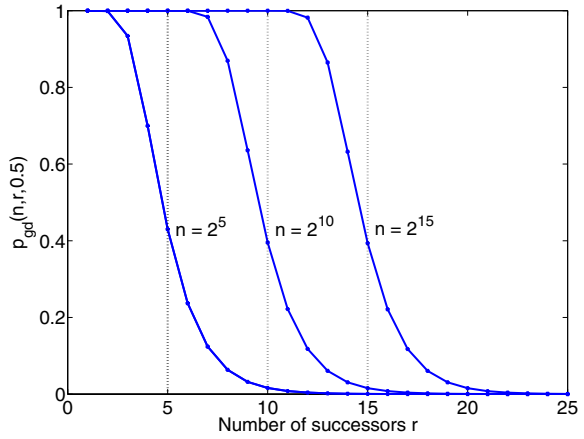

Fig. 10. Probability of a global disconnection for different successor list sizes.

Note that so far we calculated disconnection probabilities within one single *stabilize()* period. As mentioned in Section IV, the probability of a global disconnection increases over time. The longer the Chord ring exists, the greater the probability that it gets disconnected within its lifetime. Fig. 11 plots the probability that a Chord ring gets disconnected sometime within i *stabilize()* calls against the number of *stabilize()* calls for different global disconnection probabilities (cf. Eq. (8)). Assuming a *stabilize()* period of length $t_{stab} = 30$ seconds, $8 \cdot 10^4$ *stabilize()* calls roughly correspond to one month. Thus, the results demonstrate that the probability that a Chord ring gets disconnected sometime within the first month of its lifetime is by magnitudes greater than the same probability within one single *stabilize()* period.

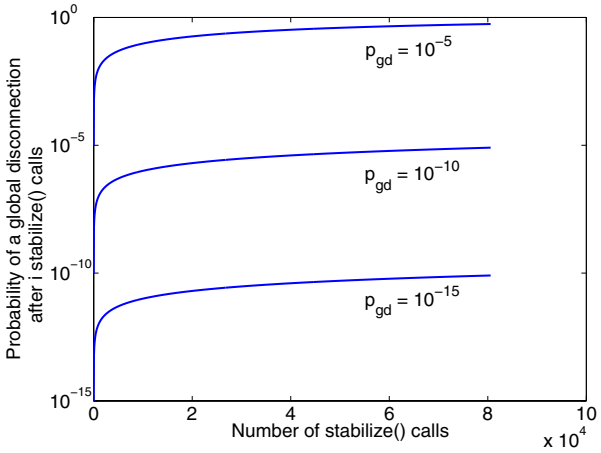

Fig. 11. Probability of a global disconnection after i stabilize() calls

VI. Conclusions

In this paper we studied the efficiency and stability of a P2P system based on the Chord algorithm. The main focus of this paper is on the problem of a disconnection of the Chord ring. In contrast to previous work it was shown, that when every peer fails with probability $\frac{1}{2}$, a successor list of size $r = \Omega(\log_2(n))$ is not sufficient to guarantee a stable Chord ring with high probability. In fact, the probability of a global disconnection is approximately 40 percent in this case.

For realistic use cases we derived an equation to calculate failure probabilities in dependence of the average online time of a peer and showed that subject to these circumstances Chord can still guarantee a stable overlay network with high probability. For system dimensioning purposes the analysis can be used to compute the actually necessary number of successors to guarantee a stable overlay network.

Acknowledgements

The authors would like to thank Prof. Phuoc Tran-Gia, Kenji Leibnitz, and Gerald Kunzmann for the help and discussions during the course of this work.

References

[1] T. Hoßfeld, K. Leibnitz, R. Pries, K. Tutschku, P. Tran-Gia, and K. Pawlikowski, "Information Diffusion in eDonkey Filesharing Networks," in *ATNAC 2004*, Sydney, Australia, December 2004.

[2] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the Evolution of PeertoPeer Systems," in *ACM PODC*, Monterey, CA, USA, July 2002.

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

[4] 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.

[5] A. Binzenhöfer, D. Staehle, and R. Henjes, "On the Fly Estimation of the Peer Population in a Chord-based P2P System," in *19th International Teletraffic Congress (ITC19)*, Beijing, China, September 2005.