THE IMPACT OF CONTENT DISTRIBUTION ON STRUCTURED P2P NETWORKS IN MOBILE SCENARIOS

S. Zöls, R. Schollmeier, Q. Hofstätter
Lehrstuhl für Kommunikationsnetze, Technische Universität München
Arcisstr. 21, D-80333 München, Germany
stefan.zoels@tum.de, ruediger.schollmeier@tum.de, q@mytum.de

A. Tarlano, W. Kellerer
Future Networking Lab, DoCoMo Communications Laboratories Europe
Landsberger Str. 312, D-80687 München, Germany
tarlano@docomolab-euro.com, kellerer@docomolab-euro.com

Abstract: Structured Peer-to-Peer (P2P) networks are promising approaches for the use of P2P technologies in resource constrained environments such as mobile communication scenarios. The signaling overhead can be reduced significantly by using a hash function to map the nodes as well as the shared objects onto the same identifier space. Queries need not be flooded through the network but can be routed directly to the responsible node. This node stores references to all peers in the network providing the requested object. The mapping algorithm of the used protocol and the actual content distribution determine the number of object references one node is responsible for. This responsibility affects the maintenance traffic in structured P2P networks, as references (responsibilities) have to be shifted among nodes when a node joins or leaves the overlay. Especially in mobile scenarios a high amount of maintenance traffic heavily influences the system performance. Therefore the content distribution has great impact on the applicability of structured P2P approaches. In this paper, we present the results of our experimental analysis of content distribution in an existing large P2P network and its consequences for the structured P2P system Chord. Based on these results we evaluate the applicability of structured P2P networks in mobile scenarios.

1.   INTRODUCTION

To apply Peer-to-Peer (P2P) technologies for mobile communications, the implemented P2P protocol has to be able to deal with the special challenges of such scenarios, i.e.
- the mobility of the users,
- the limited storage capacity of mobile devices,
- the low access data rate of mobile devices,
- the high failure probability of mobile users (due to wireless link breaks or limited battery resources),
- the high churn rates due to the high costs of mobile data transfer.

Promising approaches addressing such requirements are structured P2P protocols like Chord [Stoica, 2001] or CAN [Ratnasamy et al., 2001]. By using hash keys for every node and also for all content in the network they allow direct localization of content. Therefore, flooding of QUERY messages can be avoided, which can reduce the signaling overhead in comparison to unstructured P2P protocols like Gnutella 0.6 [Gnutella] or Freenet [Freenet].

Depending on the mapping algorithm of the used protocol as well as the actual content that is distributed in a structured P2P network, the responsibility for storing object references is shared between the participating nodes. In the worst case, one node would be responsible for storing all object references. As stated above, mobile environments are characterized by limited resources. Therefore it is crucial to have well balanced responsibilities among the nodes. As a result it is of major interest to find out the impact of real world content distributions on the applicability of structured P2P networks in mobile scenarios.

In this work, we use the Chord protocol as an example for structured P2P protocols. Chord assigns a unique ID to every node and every shared object in the network, e.g. by hashing a node's IP address or a file name. (Within this work, we will use the term "key" for the ID of a shared object.) All nodes are ordered in a one-dimensional ring that covers an ID range from 0 to 2^m-1, where m is the length of the IDs in bit. A reference to a shared object with key k is assigned to the first node whose ID is equal to or follows k. This node is therefore called the successor of k. A detailed description of the Chord protocol is given in [Stoica et al., 2001].

Assuming that the used hash function spreads all keys evenly over the nodes, every node in the Chord ring is responsible for

$$f = \frac{K}{n} \quad \forall n$$

keys, where K is the total number of keys and n the total number of nodes in the network. Although all keys are equally distributed in the network, the number of shared objects mapped onto a key can vary significantly. For example, the successor of the hash value of "Madonna" stores significantly more object references than the successor of the hash value of "Diamond Dog", a relatively unknown German rock band. In unfavorable cases, this may lead to storage capacity problems. Furthermore, all object references have to be transferred from or to the succeeding node when a node joins or leaves the network, in order to maintain the hash key mapping rules. As we must assume high churn rates in mobile scenarios due to the movement and the short online times of mobile nodes, this can result in a heavy traffic load for the involved peers.

The goal of this paper is to evaluate the impact of content distribution on the applicability of structured P2P concepts in mobile scenarios. As there currently exists no large structured P2P network, we evaluate the content distribution of a large unstructured P2P network (Gnutella 0.6 [Gnutella]) and derive the consequences of this distribution for a Chord network in mobile scenarios. In particular we have performed the following steps:

1. Find as many shared objects as possible that are provided in the Gnutella 0.6 network.
2. Determine the distribution of the discovered content in a Chord network with a given number of nodes.
3. Investigate the consequences of the determined distribution for a mobile P2P scenario.

The remainder of this paper is structured as follows. In section 2 we give an overview about our methodology for the content analysis in Gnutella. One basic criterion is to determine the optimal measurement period for a query, which is explained in detail in section 3. In section 4 we analyze the results of our measurements. In section 5 we draw conclusions for the applicability of structured P2P protocols in mobile scenarios. Finally, in section 6 we conclude our results and provide an outlook to future work.

2. PREREQUISITES FOR THE CONTENT DISCOVERY

To discover the availability and distribution of shared objects in a real world P2P network, we create a list with 578,627 different keywords, as

described below. Additionally, we modify the Gnutella client application "Mutella" [Mutella] to automatically start a query for every keyword and to log the number of QUERY HIT messages for every query.

2.1   Creation of the keyword list

To create a list with a high number of different keywords we combine the contents of different, free accessible dictionaries:
–   the German-English-Dictionary of the Technical University of Chemnitz
    with more than 170,000 entries [TU Chemnitz]
–   the "National Puzzlers' League" 2nd unabridged dictionary [Unabridged]
    (235,544 entries)
–   the "Official Scrabble Player's Dictionary" [Scrabble] (172,823 entries)
In addition we take into account that today's content of P2P networks mainly consists of compressed audio files. As the names of most song writers and artists can not be found in a normal dictionary, we also add the content of FreeDB [FreeDB] (1,486,728 entries) to our list. FreeDB is a free music data base that contains information about artists and CDs. Combining the data of all databases mentioned above and removing duplicated entries, we finally can generate a list with 578,627 different keywords.

2.2   Modification of the Mutella client

As Gnutella 0.6 is one of today's most popular P2P protocols with an open protocol specification and open source clients, we use the Gnutella 0.6 network for our content measurements. An appropriate open source client is "Mutella", a command line based Linux client implemented in C++. To use it for our measurements the "Mutella" client is modified so that it can automatically perform the following tasks:
–   Read a given number of keywords from the list.
–   Start a query for every keyword.
–   Collect and count the received QUERY HIT messages separately for every
    keyword.
–   After a given measurement period t_q, write every keyword together with
    the number of QUERY HIT messages into a log file.
To eliminate statistical aberrations, we conduct the measurements five times within several weeks and build the average values of all results.

3.   DETERMINATION OF THE OPTIMAL MEASUREMENT PERIOD

When querying an unstructured P2P network like Gnutella 0.6, every QUERY message is flooded through the network. All peers that receive a QUERY message and share objects that match the keyword stated in the QUERY message, respond with a QUERY HIT message. Due to this search mechanism, the time between initializing a query and receiving the according QUERY HIT messages can vary significantly, according to the number of hops between the querying node and the replying node and also according to the replication rate of the requested content.

As already mentioned in section II, the time to collect the according QUERY HIT messages for one QUERY is restricted to t_q. To keep the overall time for the measurements short, the measurement period t_q for one query should be as small as possible. On the other hand, t_q must be long enough to find the majority of the peers in the network that share objects that match the keyword stated in the initial QUERY message.

To determine the optimal value for t_q, we use five common keywords and trigger the Gnutella 0.6 network with a QUERY message for every keyword. We log every five seconds the number of received QUERY HIT messages. In Figure 1, the cumulative number of received QUERY HIT messages for the keywords "easy", "madonna", "music", "contact" and "value" is plotted against the duration of the measurement period.
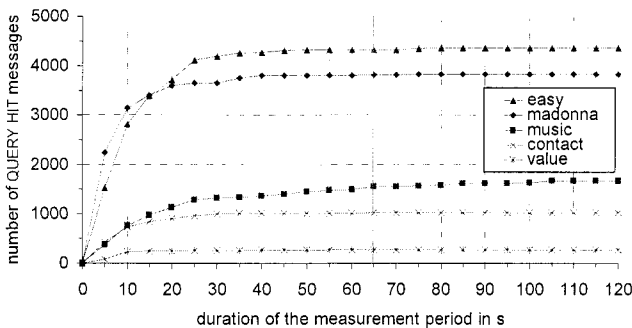

Fig.1.   Number of received QUERY HIT messages against the duration of the measurement period

As we can observe from the diagram the number of QUERY HIT messages increases significantly at the beginning of the measurement period and reaches a level of saturation after approximately 65 seconds. Afterwards, the

growth in the number of received QUERY HIT messages can nearly be neglected.

Resulting, we use in all our following measurements a measurement period of $t_q = 65$ s for every keyword. Thus we can keep the overall time for the measurements short and at the same time can guarantee that we can find a majority of the peers that share objects matching the used keyword.

4.   ANALYSIS OF THE MEASUREMENT RESULTS

We conduct our measurements five times to eliminate statistical aberrations. Table 1 shows the results of all five measurements and the average number of QUERY HIT messages. Our first measurement results in 4,183,027 QUERY HIT messages for 578,627 different keywords. The next measurements lead to 4,058,265, 4,486,803, 3,991,511 and 4,111,175 QUERY HIT messages, respectively. This yields an average number of QUERY HIT messages of 4,166,156. Throughout the rest of this paper we will use the average values of all five measurements.

Table 1. Results of all five measurements and average number of query hit messages

number of keywords	578,627
total number of QUERY HIT msg., 1st try	4,183,027
total number of QUERY HIT msg., 2nd try	4,058,265
total number of QUERY HIT msg., 3rd try	4,486,803
total number of QUERY HIT msg., 4th try	3,991,511
total number of QUERY HIT msg., 5th try	4,111,175
average number of QUERY HIT msg.	4,166,156

One result from our measurements is that 495,648 or 85.7 % of all keywords result in zero QUERY HIT messages. For the remaining keywords, we receive between one and 4,595 QUERY HIT messages for each keyword.

In Figure 2, the percentage of successful queries (= keywords that lead to at least one QUERY HIT message) with a given maximum number of QUERY HIT messages is plotted against the number of QUERY HIT messages. For example, 29.2 % of all successful queries produce just one QUERY HIT message, while 40.5 % result in one or two QUERY HIT messages. As expected, there are many keywords that result in only a few QUERY HIT messages, while only a minority of the used keywords leads to a high number of QUERY HIT messages.
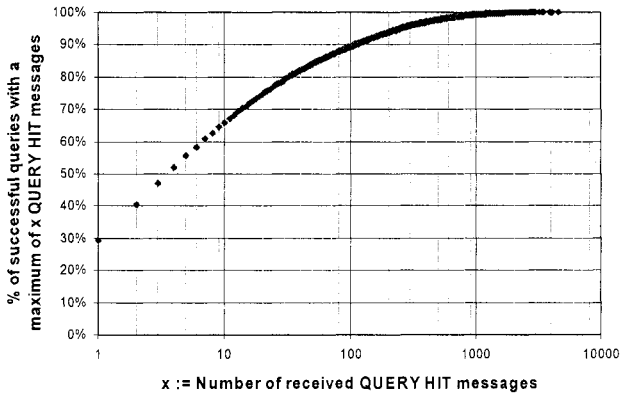
Fig. 2. Percentage of successful queries with a given maximum number of QUERY HIT
messages against the number of QUERY HIT messages

Based on our measurement results in the Gnutella network, we calculate
the resulting load balance for nodes in a structured P2P network containing
the discovered content. As stated above, we use the Chord protocol as an
example for a structured P2P network. Chord is a DHT-based protocol that
uses a hash function like SHA-1 [FIPS, 1995] to derive the keys of all shared
objects. For example, the SHA-1 key of "Madonna" is
"e4ddaa79edcafaacb559c5731edaa55455f2b069". All shared objects that
match the keyword "Madonna" are therefore mapped onto that node that is
responsible for this key, i.e. with a minimum positive difference between the
node's hash value and the hash value of the keyword.

Assuming that all nodes are distributed uniformly over the whole ring
(this is guaranteed by the use of an appropriate hash function) all nodes are
responsible for an equally sized ID section of the Chord ring. With regard to
this consideration, we set up a fictive Chord ring, split up the ID space of
this ring into n equally sized sections and assign every section to one node (n
again is the total number of nodes in the Chord network). Thus the ID
section of node j is given by

$$I_j = \left((j-1) \cdot \frac{2^m}{n}, j \cdot \frac{2^m}{n} \right] \quad \forall j \in [1, n]$$

where m is the length of the IDs in bit. Now all QUERY HIT messages
received for a certain keyword are assigned to that node that is responsible
for the keyword.

Therefore node j has to store references to all shared objects with key k that satisfy

$$k \in I_j$$

Although the number of keys per node is equally distributed by the use of an appropriate hash function, the number of object references per node can vary significantly. Figure 3 shows the load balance that results for our fictive Chord ring ($m = 160$).
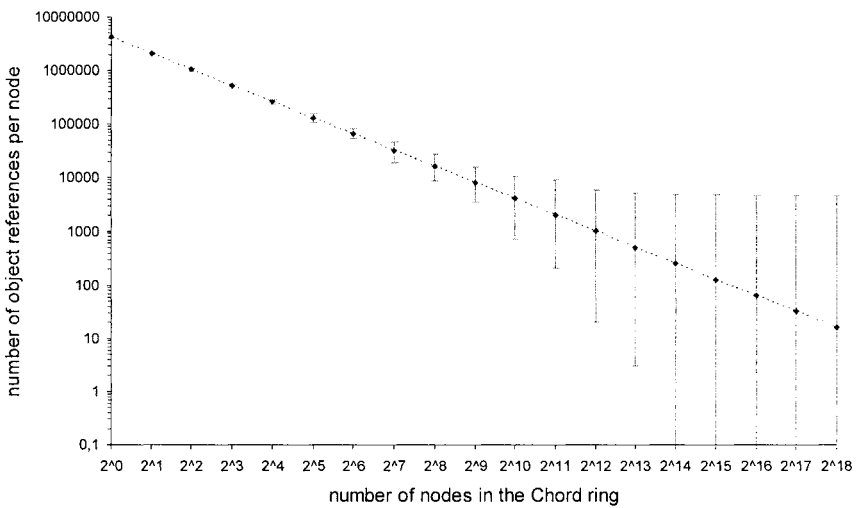

Fig. 3. Number of object references per node against the total number of nodes in the Chord ring

In the diagram, the number of object references per node is plotted against the total number of nodes participating in the Chord ring. Assuming an equal number of object references per key, any node j would store the same number of object references

$$e_j = \frac{o}{n} \quad \forall\, j \in [1; n]$$

where o denotes the total number of shared objects and n the total number of nodes in the network. e_j refers to the average value for the number of object references per node in our scenario. As Figure 3 uses a double-logarithmic

scale, e_j is depicted as a straight line. For every simulated number of nodes between 2^0 and 2^{18}, the deviation from e_j is plotted. The deviation is given by the node with the highest load and the node with the smallest load. For example, when spreading all object references over 2^{12} nodes, the average load per node is given by

$$e_j = \frac{o}{n} = \frac{4{,}166{,}156}{2^{12}} = 1{,}017$$

However, as we can see from the diagram, there are nodes whose load deviates considerably from this value. The node with the highest load is responsible for 5,870 object references, while the node with the lowest load has to store only 20 object references. Another appealing result is the fact that for a number of nodes larger than 2^{14} the highest load of a node is not decreasing any more with an increasing number of nodes in the network. This follows directly from Figure 2, as the keyword with the highest number of QUERY HIT messages (4,595) is mapped onto one node and therefore this node has to store all references to the 4,595 shared objects that match this keyword.

5.   APPLICABILITY OF CHORD IN MOBILE SCENARIOS

From the results of our measurements, we are now able to evaluate the applicability of the Chord protocol in mobile scenarios. We separate mobile communication networks into two main categories: Cellular networks and Mobile Ad Hoc Networks (MANETs).

5.1   Applicability in cellular networks

Cellular wireless networks are one-hop wireless networks. Mobile P2P nodes are thus connected to the fixed Internet only via a single wireless link. Resulting we can state that even if nodes move, the physical path to a node alters not much. Further on we can assume that most problems resulting from the mobility of the cellular wireless node are already managed by lower layers of the cellular network. We therefore have to evaluate the applicability of the Chord system in a cellular network based on the following aspects:
–   storage capacity
–   access data rate

– content availability

As derived from above, the minimum number of object references that one node always has to store is 4,595. Assuming that an object reference consists of a keyword with 20 Bytes, a hash value with 160 bit (20 Bytes), an IP address with 32 bit (4 Bytes) and a port number with 16 bit (2 Bytes), the total load of this node is given by

$$\ell = 4,595 \cdot 46 \; Bytes = 206.41 \; kB$$

This load is acceptable for the storage capacity of today's mobile devices. Even if the load of a node exceeds this value (e.g. due to a very low number of nodes in the ring) we can assume enough storage capacity on the mobile device to handle the load.

However, regarding the available access data rate of mobile devices, we have to take into account the higher churn rates of mobile nodes in cellular networks. Every time a node joins the network, all object references the node is responsible for have to be transmitted from the node's successor to the new node, in order to maintain the hash key mapping rules. In the above example, more than 200 kB have to be transferred. (On a mobile phone running GPRS with a data rate of 100 kbit/s, this would take more than 16 s!) As online time in cellular networks is expensive, the user will usually not stay in the Chord ring for a long period of time. Thus the same amount of data has to be retransferred to the node's successor after a short period of time when the node leaves the network again. For this reason, the limited access data rate of mobile devices can be a significant challenge for structured P2P protocols like Chord. It becomes even more critical when the protocol has to deal with high churn rates as they usually appear in mobile scenarios. In this case, a high amount of the available data rate is consumed by network maintenance traffic. (As we have seen above, a joining or leaving node may be blocked for 16 seconds from any further communication.)

Another critical aspect when using structured P2P protocols in a cellular network is the comparatively high probability for a connection loss. As mobile devices have commonly a wireless connection to the Internet and only a limited power supply, the probability that a mobile Chord node fails is notably higher than in a fixed environment. When a Chord node fails, all its object references are lost, if they are not replicated on other nodes. A failed node has no possibility to transfer its object references to the node's successor. This results in the unavailability of all content that is mapped onto the failed node, until the content is republished by the sharing nodes and therefore available again. The high variance of content distribution, as depicted above, makes this problem even more evident.

Concluding, we can state that structured P2P networks are, generally speaking, a suitable solution for cellular networks, because the signaling traffic in structured P2P networks is lower than in unstructured approaches. However, the two main challenges we still have to bear in mind are the low transmission data rate of mobile devices and the high probability of a connection loss.

5.2   Applicability in MANETs

If we want to operate a structured Peer-to-Peer network on top of a MANET, we have to take into account the mobility of the nodes. The reason therefore is that MANETs are self configuring, wireless multi-hop networks, within which we have to assume the possibility that the physical path between two nodes changes frequently, due to the movement of the nodes [Gruber and Li, 2003]. Besides the challenges mentioned above, i.e. low transmission data rates and high failure probability, further problems can occur in MANETs:

If the overlay structure is established completely independent from the underlying MANET topology, this results in a significant number of long and unstable zigzag routes in the MANET layer. The connections between the overlay nodes can not be mapped onto the physical topology of the MANET [Eberspächer et al. 2004]. This leads to significantly higher traffic volumes, which might not be bearable by the MANET [Klemm et al. 2003]. Further on we have to take into account the significant overhead to keep alive overlay connections via more than 5 physical hops. According to [Gruber and Li, 2003] paths longer than 5 hops hardly survive more than a few seconds.

However, if we assign IDs to the participating overlay nodes which reflect the current position of a node in the MANET, an overlay network is established that solves the problems mentioned above, i.e. zigzag routes and long physical paths. As the IDs reflect the current position of a node, as proposed in [Zhuang and Zhou, 2003], the connection between two overlay nodes with a small distance in the ID space also results in a small distance in the physical space of the MANET. However, as the nodes are assumed to move, the IDs of the nodes have to change constantly, to reflect the changing physical network. This results in a continuous transfer of object references between the nodes, as the IDs and therefore the responsibility of the nodes changes constantly. Concluding, this results in high network maintenance traffic, especially when the content distribution has such a high variance as pointed out in section IV.

With regard to the above conclusions we can state that structured P2P networks are hardly applicable in MANETs. For the use of P2P technologies

in such networks, other promising protocols that are based on an unstructured overlay architecture, like the Mobile Peer-to-Peer protocol [Schollmeier et al., 2003] are more suitable.

6.   CONCLUSION AND FUTURE WORK

In this paper, we presented the results of our experimental analysis of content distribution in an existing Gnutella 0.6 network and derived the consequences this distribution would have for the structured P2P system Chord. Based on these results we evaluated the applicability of structured P2P networks in mobile scenarios.

To find approximately all content that is shared in the Gnutella network, we created a list with a high number of keywords. Then we queried the network for every keyword and logged the number of QUERY HIT messages after a given measurement period t_q. As illustrated in section III, the optimal measurement period in the Gnutella network is 65 s. The results of our measurements show that many keywords result in no or only a few QUERY HIT messages and only several keywords lead to a high number of QUERY HIT messages.

In a second step, we apply our measurement results on a fictive Chord system with a given number of nodes. For this case we determine the load balance that results in the Chord ring. With an increasing number of nodes, the deviation from the average number of object references per node e_j increases significantly.

Based on these results, we can evaluate the applicability of structured P2P protocols like Chord in mobile scenarios, where we differentiate between cellular networks and Mobile Ad Hoc Networks. In cellular networks, structured P2P networks can be applicable, although the low transmission data rate of mobile devices and the high probability of a connection loss are challenges that have to be met. In MANETs, structured P2P networks are hardly useable. For MANETS, other promising P2P protocols, based on an unstructured architecture, have been proposed.

Currently we are working on a modification of the Chord protocol that aims to reduce the necessity of re-transmission of object references between low performance nodes [Zöls et al., 2005]. Thus the necessary maintenance traffic can be reduced significantly and adaptations to the physical network are possible. Our approach defines a two-tier architecture with two different classes of nodes participating in the Chord network. Firstly, we have a class of static, highly available nodes with higher capabilities, like a broadband connection. These nodes store the references to all objects that are provided in the Chord ring. Secondly, we have temporary nodes like mobile

subscribers that do not store object references but forward all requests to their next static successor. Thus we can employ the advantages of structured P2P networks even in mobile scenarios.

REFERENCES

Eberspächer, J., Schollmeier, R., Zöls, S. and Kunzmann, G. (2004). Structured Peer-to-Peer Networks in Heterogeneous Environments. In Second International Conference on Heterogeneous Networks (Hetnets'04).

FIPS (1995). "Secure Hash Standard" Federal Information Processing Standards Publication FIPS PUB 180-1.

FreeDB, Database to Look Up CD Information Using the Internet, http://www.freedb.org

Freenet, Freenet Homepage, http://freenetproject.org/cgi-bin/twiki/view/Main/WebHome

Gnutella 0.6, RFC-Gnutella 0.6, http://rfc-gnutella.sourceforge.net/developer/testing/

Gruber, I. and Li, H. (2003). Path Expiration Times in Mobile Ad Hoc Networks. Presented at European Personal Mobile Communications Conference (EPMCC'03), 2003.

Klemm, A., Lindemann, C. and Waldhorst, O. P. (2003). A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks. Presented at Workshop on Mobile Ad Hoc Networking and Computing (MADNET 2003).

Mutella, Mutella Homepage, http://mutella.sourceforge.net

Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. (2001). A Scalable Content-Addressable Network, presented at ACM SIGCOMM Conference.

Schollmeier, R. Gruber, I. and Niethammer, F. (2003). Protocol for Peer-to-Peer Networking in Mobile Environments in International Conference on Computer Communications (ICCCN03).

Scrabble. The National Puzzlers' League, "The Word Lists: The Official Scrab-ble™ Player's Dictionary", ftp://puzzlers.org/pub/wordlists/enable1.txt

Stoica, I., Morris, R., Karger, D., Kaashoek, M. and Balakrishnan, H. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In ACM SIG-COMM Conference.

TU Chemnitz, "Deutsch-Englisches Wörterbuch", http://dict.tu-chemnitz.de

Unabridged. The National Puzzlers' League, "The Word Lists: 2nd Unabridged Dictionary", ftp://puzzlers.org/pub/wordlists/unabr2.txt

Zhuang, L. and Zhou, F. (2003). Understanding Chord Performance. Technical Report CS268.

Zöls, S., Schollmeier, R., Kellerer, W. and Tarlano, A. (2005). The Hybrid Chord Protocol: A Peer-to-Peer Lookup Service for Context-Aware Mobile Applications, Internat. Conf. on Networking (ICN'05).