Determining the Hop Count in Kademlia-type Systems

Stefanie Roos*, Hani Salah†, Thorsten Strufe*
*TU Dresden, Mommsenstr. 8, 01061 Dresden, Germany
firstname.lastname@tu-dresden.de, Tel. +49 (0) 351 463-38448
†TU Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
hsalah@ca.tu-darmstadt.de, Tel. +49 (6151) 16 - 23206

***Abstract*—The family of Kademlia-type systems represents the most efficient and most widely deployed class of Internet-scale distributed systems. However, prior research on these systems has mainly been restricted to analyzing deployed systems and suggesting improvements tailored to specific environments rather than exploiting the huge parameter space governing the routing performance. Concise analytic results are rare, due to the complexity of Kademlia's parallel and non-deterministic lookups.**

This paper introduces the first comprehensive formal model of the routing for the entire family of Kademlia-type systems. We validate our model against simulations of both the BitTorrent Mainline DHT and eMule's KAD implementation. The model allows a highly scalable comparison with respect to the hop distribution of different variations to the original protocol. In particular, we show that several of the recent improvements to the protocol in fact have been counterproductive with regard to routing efficiency.

***Index Terms*—Kademlia, DHTs, Routing, Markov Chain**

I. Introduction

Kademlia and its variations represent the class of the most deployed and most actively used large scale distributed discovery services, today. They allow to map objects in the form of key-value pairs to nodes. To resolve this mapping, they facilitate a decentralized message routing to the respective node in a very robust fashion. This resolution is highly efficient in that the state every node needs to store as well as the number of nodes that are contacted to resolve a discovery request grow only logarithmically in the number of participants.

The novel idea of Kademlia is to implement an iterative, greedy routing over parallel paths using the XOR of compared identifiers as the distance metric and to adopt a replication factor for the storage of objects and routing state. It has been implemented in several subsequent variations that adapt the routing and the replication parameters. The current BitTorrent clients span a Kademlia-type overlay for node discovery, with up to 27 million concurrent users [1]. Another popular example is the Kademlia derivative KAD that has been integrated into the eDonkey file-sharing network to discover sources of file chunks and has grown to deployments of more than a million concurrent nodes [2].

Despite the considerable attention Kademlia received from both research and industry, the impact of the design parameters on the routing performance is poorly understood. Measurements only offer insights on deployed systems, whereas simulations do not scale beyond a few tens of thousands nodes. Existing analytical results fail to include important parameters, network dynamics, and are mostly only asymptotic bounds. To assess the effect of chosen parameters covering the entire design space, we provide an analytical model that is formalized as a Markov chain. Our approach differs from classical theoretical analysis as that concrete bounds are provided for certain parameters rather than a general complexity analysis.

We thus obtain an analytic framework that is similarly flexible and precise as simulations, but much more scalable. In contrast to previous hop count estimations, our model incorporates all essential parameters governing Kademlia-type systems. In particular, we manage to integrate parallelism, which the most widely known model by Stutzbach and Rejaie fails to consider [3]. It is highly efficient in storage and computation, allowing to easily analyze systems of up to a billion nodes.

We validate the model against results from simulations, showing close agreement within a 95% confidence interval. Analyzing the parameter space allows us to give guidelines for future design adaptations. With respect to the different suggested modifications, we show that they frequently do not outperform the original proposal for networks of several millions of nodes. For example, the changed degree of parallelism in BitTorrent provides a higher average hop count than the original parameters for networks of more than one million nodes.

We present our model of the hop count in Section IV, after stating our notion of a Kademlia-type system in Section III. The storage and computation complexity of the model is analyzed in Section V. In Section VI, we validate the model, show its scalability, and present an exemplary parameter study indicating its advantage over simulation studies.

II. Kademlia-type systems

In this section, we give a short overview of the concepts Kademlia is based on, before presenting various studies on modeling and analyzing P2P routing, with a focus on Kademlia.

A. Introducing Kademlia

Kademlia [4] is a structured peer-to-peer (P2P) system. Nodes and objects are assigned IDs from the same b-bit ID space and the distance between two IDs is defined as the XOR

of their values. Kademlia implements key-based routing and storage of key-value (ID-object) pairs. The nodes at the closest distance to an object's ID are responsible for storing it.

Each node v maintains a routing table to store the IDs and addresses of other nodes, without keeping persistent network connections to them. In Kademlia, the neighbors, also called *contacts*, are stored in a tree-like routing table structure. The level in the tree a contact w of v is stored at reflects the common prefix length of v and w. At most k contacts are stored at each level, making up a so called *k-bucket*. Information stored in the routing table may be outdated, or *stale*, when the respective nodes have left the system.

Kademlia implements greedy routing: To route a message from node v to a target ID t (for the storage or retrieval of objects), v sends parallel lookup requests to the α known contacts that are closest to t. Every queried contact that is online replies with the set of β nodes that are locally known as being closest to t, thus extending v's set of candidate contacts. This process is iterated until the lookup does not produce any contacts closer to t than previously have been discovered, or a timeout is caught. The original Kademlia publication suggests to use $k = 20$ and $\alpha = 3$.

Kademlia proved highly efficient and reliable, and thus has frequently been modified, generating a broad family of Kademlia-type systems. Each adaptation mainly adjusts the given parameters, or the routing table structure. The current mainline implementation of BitTorrent (MDHT), for example, integrates a Kademlia-type DHT for node discovery. uTorrent, the most popular client implementing MDHT, is implemented using 8-buckets, $\alpha = 4$, and $\beta = 1$ [5]. To reflect the fractions of the ID space that are covered at different levels, and hence to increase the distance reduction at each hop, variable bucket sizes k_i are introduced in iMDHT [5]. They are chosen to be 128, 64, 32, and 16 for the buckets at levels $i \in (0..3)$ respectively, and 8 for all lower levels. The variation used in the highly popular eDonkey file-sharing network, KAD, adds multiple buckets per level, grouping contacts according to the first l bits after the first diverging bit. This way, the *bit gain*, i.e. the difference between the common prefix length of the current hop and the next hop to t, is at least l. Choosing k to be 10, the implementation contains buckets for all 4-bit prefixes at level 0 (containing contacts that share no common prefix with v), and one bucket for each of the sub-prefixes 111, 110, 101, 1001, and 1000 at all remaining levels. Thus, the guaranteed distance reduction is 3 bits for 75% of the targets IDs, and 4 bits for the remaining 25%. By default, KAD implements $\alpha = 3$ and $\beta = 2$.

B. Analyzing P2P Routing

Motivated by the success and popularity of Kademlia-type systems, a large number of studies over the past few years [2], [5]–[9] focused on these systems. These studies, however, are mainly based on large-scale measurements, and do not yield insight into the impact of isolated design adaptations. Analytic results of the routing are rare. Existing studies are largely restricted to the asymptotic worst-case complexity of $\mathcal{O}(\log n)$ routing steps for a network of order n. A notable exception is [3], in which a formula for the average hop count is derived. This derivation, however, considers only the KAD implementation and fails to give further insight into the hop count distribution. It hence does not allow for the choice of sensible timeout duration and termination criteria for more sophisticated, possibly time-critical, applications. Parallel lookups and variations in the routing table structure also are disregarded.

In this paper, we model Kademlia-like systems as a stochastic process in agreement with influential works on distributed routing such as [10]–[12]. However, they provide asymptotic bounds rather than concrete results, so that they do not allow the comparison of variants with the same asymptotic complexity. Some of the few works deriving numerical bounds are [13], [14], which are very close to our own approach. Both model overlay routing as a Markov chain. The state of the Markov chain represents the distance of the currently contacted node's ID to the target ID. The expected hop count is then derived as the expected number of steps until the Markov chain reaches the absorbing state, which corresponds to discovering the target. However, these results do not consider parallelism and the hop distribution, and use simplified assumptions such as a bijective mapping from nodes to IDs. In addition to modeling a considerably more complex system of practical importance, our work also addresses the question of minimizing the storage and computation complexity of the model. These aspects are disregarded in the related work, due to the low number of states needed to characterize non-parallel lookups.

III. Notation

In this section, we first introduce the concept of Markov chains. Afterwards, we formalize our definition of a Kademlia-type system.

A. Markov Chains

A *Markov chain* is a *random process* $X_0, X_1, \ldots$, such that the probability distribution of X_{i+1} only depends on X_i. Formally, a random process with *state space* S is a Markov chain if

$$\begin{aligned}\forall x_0, \ldots, x_{i+1} \in S : &P(X_{i+1} = x_{i+1} | X_i = x_i, \ldots, X_0 = x_0) \\ &= P(X_{i+1} = x_{i+1} | X_i = x_i).\end{aligned} \tag{1}$$

It follows from Eq. 1 that Markov chains are memoryless, i.e.

$$P(X_{i+1} = x_{i+1} | X_i = x_i) = P(X_1 = x_{i+1} | X_0 = x_i). \tag{2}$$

By Eq. 2, the probability distribution X_i can be obtained by straightforward matrix multiplication if the state space S is finite and the distribution X_0 is known. Then X_i can be obtained by matrix multiplication as follows: The finite set S can be enumerated as $S = \{s_0, s_1, \ldots, s_{|S|-1}\}$. The *initial distribution* of the Markov chain is given by a $|S|$-dimensional vector I, such that the j-th entry $I(j)$ is $I(j) = P(X_0 = s_j)$. Similarly, the probabilities $P(X_1 = x_1 | X_0 = x_0)$ are mapped to entries of a $|S| \times |S|$ *transition matrix* T. The entry $T(i, j)$ is

$T(i,j) = P(X_1 = s_i | X_0 = s_j)$. The probability distribution of X_{i+1} is hence obtained as

$$X_{i+1} = T^{i+1} I. \tag{3}$$

In this paper, we will use the distances of the closest α contacts to the target ID as states.

B. Model Overview

The common prefix lengths of the closest nodes to the target ID t is used to characterize the routing process. Because the decisive factor in Kademlia routing is the common bit length, we define the distance of two nodes w and v to be the bit length of the XOR of their IDs

$$\begin{aligned} dist(w,v) &= b - commonprefixlength(id(w), id(v)) \\ &= \lfloor \log_2 XOR(id(w), id(v)) \rfloor + 1, \end{aligned} \tag{4}$$

where b is the identifier space size and $id(v)$ denotes the b-bit ID of node v. We here use distance to refer to $dist$ rather than the XOR distance, unless stated otherwise.

We formally characterize a Kademlia-type system by the ID space size b, the routing parameters α and β, and the routing table parameters k and L. L determines the number of buckets per level as well as how the ID space is split among these buckets.

Definition III.1. *A $\mathcal{K}(b, \alpha, \beta, k, L)$-system is a Kademlia-type system with the following properties:*

- *A b-bit ID space is used for addressing.*
- *α parallel iterative queries are sent for each lookup.*
- *Each queried node answers with at most β contacts closer to the target than itself.*
- *The d-entry k_d of the vector $k \in \mathbb{N}_0^{b+1}$ gives the bucket size for nodes with distance d to the routing table owner (i.e. the bucket size at level $b - d$).*
- *The i-th row of the matrix L gives the distribution of the guaranteed bit gain at distance i to the routing table owner, i.e. the entry $L_{ij} = \frac{x}{2^i}$ is defined by the number x of IDs with distance i that are sorted in buckets covering a region of 2^{i-j} IDs each.*

Furthermore, the network order n influences the hop count distribution. Note that in most Kademlia-type systems, such as MDHT as well as KAD, k is constant. Similarly, the matrix L is commonly sparse. For instance, in MDHT only one bucket is used for each common prefix length, so $L_{i1} = 1$ for $i = 0 \dots b$ and $L_{ij} = 0$ in all other cases. KAD is more complicated: $L_{b4}^{KAD} = 1$, $L_{i3}^{KAD} = 0.75$, and $L_{i4}^{KAD} = 0.25$ for $i < b$ determine the routing table structure in the KAD system. This is due to resolving at least 4 more bits on the top level, and splitting into buckets with prefixes 111, 110, 101 (75% of IDs within that bucket), as well as 1001 and 1000 (25% of IDs) for all lower levels.

In addition to the above notation, we use $B(n,p)$ to denote a binomial distribution with n trials and success probability p.

b	Bit-length of IDs
α	Degree of parallelism
β	Number of returned contacts
k_i	Bucket size on level i
L_{ij}	Fraction of buckets with 2^{i-j} IDs on level i
S_α	State space of distance sets
I	Initial distribution
T	Transition matrix
T^{up}/T^{low}	Transition matrix upper/lower bound
X_l	l-th state of Markov chain
D	Distance distribution
γ	Denotes either α or β
C_γ	Distribution of closest γ neighors
$\mathbb{B}(n,p)$	Binomial distribution with parameters n, p
$F_{d,l}$	Distribution of random ID with distance $d - l$

TABLE I: Important Notation

IV. Deriving the Hop Distribution

The goal is to obtain a close approximation of the hop distribution in a $\mathcal{K}(b, \alpha, \beta, k, L)$-system, i.e. the probability distribution of hops needed to route a query from a requesting node v to a target node t. We approximate the routing process as a Markov chain. Let $\emptyset$ denote the fact that the target node is found and the routing is terminated. If the target node has not been found, we describe the state of the routing by the distances of the α currently contacted nodes. Formally, the routing state at hop X_i is an element of the state space

$$S_\alpha = \{\emptyset\} \cup \{(z_1, \dots, z_\alpha) : z_j \in \mathbb{Z}_{b+1}, z_j \leq z_{j+1}\}.$$

In order to describe the routing by a Markov chain, we assume that the main determining factor for X_{i+1} is X_i, i.e.

$$\begin{aligned} &P\left(X_{i+1} = s_{i+1} | X_i = s_i, \dots, X_0 = s_0\right) \\ &\approx P\left(X_{i+1} = s_{i+1} | X_i = s_i\right). \end{aligned} \tag{5}$$

The assumption is based on the observation that the new contacts are chosen from the routing tables of the α currently contacted nodes, which are chosen independently. All earlier states seem to have a negligible influence. They provide information on how many nodes have been contacted and hence slightly reduce the number of nodes that can be contacted, but due to the shortness of routes, we assume this change to have a negligible impact. Furthermore, knowing that a certain progress was made or not made in earlier steps hints on the distribution of nodes over the identifier space, e.g. if nodes with a certain prefix are contacted more often than expected, the number of nodes with such a prefix is likely to be higher than expected. Again, we deem this factor to be negligible because nodes contacted during routing represent a very small sample of nodes.

We hence analyze the Markov chain corresponding to the right side of Eq. 5. First, we determine the initial distribution I of the distances of the closest nodes to t in the requesting node's routing table. Secondly, we obtain transition matrices T^{up} and T^{low}. So, concrete upper and lower bounds on the probability that the above Markov chain reaches the absorbing state $\emptyset$ in i steps are given by $(T^{up})^{i-1} I$ and $(T^{low})^{i-1} I$. The main difficulty lies in deriving the probability distribution C_γ of the closest $\gamma \in \{\alpha, \beta\}$ contacts to t in a node u's routing table, conditioned on the distance D of u to t. The initial

distribution then follows immediately. For the derivation of the transition matrices, an additional step is needed: When deciding on a new set of nodes to query, the requesting node chooses the α closest *distinct* nodes from the $\alpha\beta$ replies received in the last hop. Upper and lower bounds are needed to cover the unlikely case of less than α distinct contacts. In the following, we start by stating our assumptions before deriving the initial distribution and the transition matrices in terms of the closest contacts C_γ. C_γ is then derived in the last part of the section.

A. Assumptions

We determine the hop distribution based on the following assumptions:

1) There are no stale contacts in the routing tables.
2) Nodes do not fail nor do they drop messages.
3) Buckets are maximally full, i.e. if a bucket contains $k_1 < k_d$ values, there are exactly those k_1 nodes in the region the bucket is responsible for.
4) Node IDs are chosen uniformly and independently.
5) Routing table entries are chosen independently.
6) If the distance between two nodes is 0, they are contained in each other's routing tables.
7) The lookup is blocking, i.e. a node awaits all answers to its α queries before sending additional ones.

Assumptions 1, 2, and 3 can be summarized as the assumption of a steady-state system, without churn or failures. However, we present an extended version of the model which incorporates churn and failure in our technical report [15].

B. Initial Distribution I

Theorem IV.1. *The probability $I(s) = P(X_0 = s)$ of the initial distribution attaining state $s \in S_\alpha$ is*

$$I(s) = \sum_{d=0}^{b} P\left(C_\alpha = s | D = d\right) \frac{2^{\max\{d-1,0\}}}{2^b}. \tag{6}$$

Proof: We sum over all possible distances d of the requesting node, so that

$$I(s) = \sum_{d=0}^{b} P\left(C_\alpha = s | D = d\right) P\left(D = d\right).$$

The probability $P(D = d)$ corresponds to the fraction of IDs at distance d. ∎

C. Transition Matrix T

Let X_0 denote the current state of the routing, and X_1^{up} and X_1^{low} the next state. We use $*$ to denote either up or low. Furthermore, let C_β^i for $i = 1 \dots \alpha$ denote the C_β-distributed bit distances of the β closest contacts to t in the routing table of the i-th contacted node in the current step. We state the transition probability in Theorem IV.2 in terms of $P\left(X_1^* = s_1 | C_\beta^1 = z^1, \dots C_\beta^\alpha = z^\alpha, X_0 = s_0\right)$.

Theorem IV.2. *The probability that X_1^* attains state $s_1 \in S_\alpha$ given $X_0 = \emptyset$ is*

$$P\left(X_1^* = s_1 | X_0 = \emptyset\right) = \begin{cases} 1, & s_1 = \emptyset \\ 0, & s_1 \neq \emptyset \end{cases}. \tag{7}$$

If $X_0 = s_0 = (d_1, \dots, d_\alpha) \neq \emptyset$, the probability of $X_1^* = s_1 = \emptyset$ is

$$\begin{aligned} &P\left(X_1^* = \emptyset | X_0 = (d_1, \dots, d_\alpha)\right) = \\ &1 - \prod_{i=1}^{\alpha} \left(1 - P\left(C_\beta = \emptyset | D = d_i\right)\right) \end{aligned} \tag{8}$$

and the probability of $X_1^* = s_1 = (\delta_1, \dots, \delta_\alpha)$ is

$$\begin{aligned} &P\left(X_1^* = s_1 | X_0 = (d_1, \dots, d_\alpha)\right) \\ &= \sum_{z_1, \dots, z_\alpha \in S_\beta \setminus \{\emptyset\}} \Bigg(\prod_{i=1}^{\alpha} P\left(C_\beta = z_i | D = d_i\right) \\ &\quad \cdot P\left(X_1^* = s_1 | C_\beta^1 = z^1, \dots, C_\beta^\alpha = z^\alpha, X_0 = s_0\right) \Bigg) \end{aligned} \tag{9}$$

Proof: Eq. 7 holds because $\emptyset$ is an absorbing state. Eq. 8 follows from Assumption 5 and gives the probability that at least one of the α considered routing tables contains t. Eq. 9 is obtained by conditioning on all possible states of the closest entries in the α considered routing tables. ∎

It remains to evaluate the term $P\left(X_1^* = s_1 | C_\beta^1 = z_1^1, \dots C_\beta^\alpha = z^\alpha, X_0 = s_0\right)$ in Eq. 9. Since we did not find a closed form, we here describe its iterative computation. In the following, we abbreviate the event $E = \{C_\beta^1 = z_1^1, \dots C_\beta^\alpha = z^\alpha, X_0 = s_0\}$. Due to space constraints, we here present the straightforward computation and refer to our technical report [15] for a more efficient solution used to produce our results in Section VI.

Let z_j^i denote the j-th element of z^i and $\rho^*(z^1, \dots, z^\alpha)$ denote the distances of the $\alpha\beta$ closest distinct nodes that have not be contacted. If not all returned contacts are distinct, replacement contacts known from earlier steps or the routing table of the requesting node are included. Due to the Markov property, we can only determine upper and lower bounds on the distance of replacement contacts. All known but not contacted nodes have distance at least d_α, so that for an upper bound on the success probability, we minimize the distance of a replacement contact by $K^{up} = d_\alpha$. In contrast, for a lower bound on the success probability, $K^{low} = b$ is chosen, corresponding to a replacement node at maximal distance to t.

Denote the set of all possible distances of distinct contacts given $z = (z_1^1, \dots, z_\beta^\alpha)$ by $U^*(z) = \{u = (u_1^1, \dots, u_\beta^\alpha) : u_j^i \in \{z_j^i, K^*\}\}$. The function $min_\alpha : \mathbb{Z}_{b+1}^{\alpha\beta} \to S_\alpha$ determines the state corresponding to the α smallest values in a $\alpha\beta$-dimensional vector. Then the desired probability is given by $P\left(X_1^* = s_1 | E\right) = \sum_{u \in U^*(z) : min_\alpha(u) = s_1} P\left(\rho^*(z) = u | E\right)$. We obtain $P\left(\rho^*(z) = u | E\right)$ for $u = (u_1^1, \dots, u_\beta^\alpha)$ iteratively,

with $\rho^*_{i,j}$ denoting the i,j-th element in $\rho^*(z)$, i.e.

$$P\left(\rho^*(z) = (u^1_1, \dots, u^\alpha_\beta)|E\right) \tag{10}$$
$$= \prod_{i=1}^{\alpha}\prod_{j=1}^{\beta} P\big(\rho^*_{i,j}(z) = u^i_j | E, \rho^*_{1,1} = u^1_1, \dots, \rho^*_{i,j-1} = u^i_{j-1}\big).$$

For brevity, we set $A_{i,j} = \{\rho^*_{1,1} = u^1_1, \dots \rho^*_{1,\beta} = u^1_\beta, \dots \rho^*_{i,j-1} = u^i_{j-1}\}$. The probability that a node at distance z^i_j is identical to one that has earlier been contacted is equal to the ratio of contacted nodes and all nodes at distance z^i_j.

We now compute $count^*\left(z^i_j, |E, A_{i,j}\right)$, the number of nodes at distance z^i_j that have been contacted and are potentially identical to the currently considered node. Note that contacts returned by the same node are distinct, so in particular they cannot be equal to the same earlier returned contact. Thus the number of contacts from the current set of returned contacts potentially identical to the j-th returned contact of queried node i is $\#z^i_j = |\{(\eta,\mu) : z^i_j = z^\eta_\mu = u^\eta_\mu, \eta < i\}| - |\{\mu : z^i_j = z^i_\mu = u^i_\mu, \mu < j\}|$, if that number is non-negative and 0 otherwise. Only at this point, we distinguish upper and lower bounds. For the upper bound, only nodes from the current set of returned contacts are considered, so that

$$count^{up}\left(z^i_j, |E, A_{i,j}\right) = \max\{0, \#z^i_j\}. \tag{11}$$

For the lower bound, the above identity holds if $z^i_j < d_1$ because no node closer than d_1 has been considered before. If, on the other hand, $z^i_j \geq d_1$, contacts at distance z^i_j might have been considered in earlier steps. Routing takes at most b steps because an improvement of one bit per step is guaranteed. In each step α nodes are contacted, so that at most $b\alpha$ nodes are contacted, giving a bound of

$$count^{low}\left(z^i_j, |E, A_{i,j}\right) = \begin{cases} \max\{0, \#z^i_j\}, & z^i_j < d_1 \\ b\alpha, & z^i_j \geq d_1 \end{cases}. \tag{12}$$

If $count^*\left(z^i_j, |E, A_{i,j}\right) = 0$, then

$$P\left(\rho^*_{i,j}(z^1, \dots, z^\alpha) = u^i_j | E, A_{i,j}\right) = 1. \tag{13}$$

Otherwise, the number of nodes Y^j_i at distance z^i_j that have not been contacted is $B(n - \alpha\beta, 2^{z^i_j - 1 - b})$ distributed, so that

$$\begin{aligned} &P\left(\rho^*_{i,j}(z^1, \dots, z^\alpha) = u^i_j | E, A_{i,j}\right) \\ &= \sum_{m=0}^{n-\alpha\beta} P\left(Y^j_i = m\right) \frac{m}{m + count^*\left(z^i_j, |E, A_{i,j}\right)} \\ &= \sum_{m=0}^{n-\alpha\beta} \binom{n-\alpha\beta}{m} \left(2^{z^i_j - 1 - b}\right)^m \left(1 - 2^{z^i_j - 1 - b}\right)^{n-\alpha\beta-m} \\ &\qquad \frac{m}{m + count^*\left(z^i_j, |E, A_{i,j}\right)}. \end{aligned} \tag{14}$$

Inserting Eq. 13 and Eq. 14 in Eq. 10 completes the computation of the $P\left(X^*_1 = s_1 | C^1_\beta = z^1, \dots C^\alpha_\beta = z^\alpha, X_0 = s_0\right)$ in Eq. 9.

D. Closest Contacts C_γ

We first consider the probability to discover t in Lemma IV.3 before treating non-absorbing states in Lemma IV.4.

Lemma IV.3. *Let B_l be a binomial distributed random variable with parameters $n-2$ and 2^{d-l-b}. The probability that C_γ attains state $\emptyset$ is*

$$P\left(C_\gamma = \emptyset | D = d\right) = \tag{15}$$
$$\begin{cases} 1, & d = 0 \\ \sum_{l=1}^{b} L_{dl} \sum_{m=0}^{n-2} P(B_l = m) \min\left\{1, \frac{k[d]}{m+1}\right\}, & d > 0 \end{cases}.$$

Proof: The first case holds by Assumption 6. For the second case, we condition on the number of further resolved bits, whose distribution is determined by the matrix L. If the node ID belongs to a bucket for which l further bits are resolved, there are 2^{d-l} IDs with the same prefix. By Assumption 4, the number of nodes with that bucket prefix is $\mathbb{B}(n-2, 2^{d-l-b})$ distributed ($n-2$ because t and the requesting node are not considered). By Assumption 3, t is contained in the bucket if there are less than k_d potential entries, otherwise the probability that t is contained in the bucket is given by ratio of the bucket size and the number of nodes with the same prefix. ∎

We introduce some notation for Lemma IV.4: First, let $F_{d,l}(x) = \min\{1, \frac{2^{\lfloor x \rfloor}}{2^{d-l}}\}$ for $x \geq 0$ be the cumulative distribution function of the distance of a random ID within distance 2^{d-l} of t. We arrange the closest contacts into groups according to their distances. More precisely, for a state $s = (d_1, \dots, d_\gamma) \in S_\gamma \setminus \{\emptyset\}$, let $\gamma'(s) = |\{d_1, \dots, d_\gamma\}|$ be the number of distinct values in s. We transform s into $s' = (M_1, \dots, M_{\gamma'})$ for $M_i = (y_i, c_i)$ with y_i being the i-th smallest distinct value in $\{d_1, \dots, d_\gamma\}$ and c_i being the number of times y_i appears in s. Define $C_i = k_d - \sum_{j=1}^{i} c_j$ as the sum of values in s bigger or equal to $y_i + 1$. Let $M_i(0)$ and $M_i(1)$ denote the first and second element of the vector M_i, respectively. We set $y_0 = -1$ and $C_0 = k_d$ for all of the following equations to be well defined.

Lemma IV.4. *The probability that C_γ attains state $s \in S_\gamma \setminus \{\emptyset\}$ is*

$$\begin{aligned} &P\big(C_\gamma = (d_1, ..., d_\gamma) | D = d\big) = \big(1 - P\left(C_\gamma = \emptyset | D = d\right)\big) \\ &\cdot \sum_{l=1}^{b} \left(\sum_{i=1}^{\gamma'-1} \binom{C_{i-1}}{c_i} \left(F_{d,l}(y_i) - F_{d,l}(y_i - 1)\right)^{c_i} \right) \\ &\quad \cdot \Bigg(\left(1 - F_{d,l}(y_{\gamma'})\right)^{C_{\gamma'-1}} - \sum_{j=0}^{c_{\gamma'}-1} \binom{C_{\gamma'-1}}{j} \\ &\qquad \cdot \left(F_{d,l}(y_{\gamma'}) - F_{d,l}(y_{\gamma'} - 1)\right)^j \left(1 - F_{d,l}(y_{\gamma'})\right)^{C_{\gamma'-1}-j} \Bigg) L_{dl}, \end{aligned} \tag{16}$$

where $P\left(C_\gamma = \emptyset | D = d\right)$ is determined in Lemma IV.3.

Proof: In the first step, we write

$$\begin{aligned} &P\left(C_\gamma = (d_1, ..., d_\gamma) | D = d\right) \\ &= P\left(C_\gamma \neq \emptyset | D = d\right) P\left(C_\gamma = (d_1, ..., d_\gamma) | D = d, C_\gamma \neq \emptyset\right). \end{aligned}$$

In the following, we condition on $C_\gamma \neq \emptyset$, the distance d and the further resolved bits l. By Assumptions 4 and 5, the

IDs of contacts in a bucket are selected uniformly at random and independently from all IDs in that region. So, the next state of the Markov chain is then given by the closest γ of k_d randomly selected IDs within the bucket with the longest common prefix to t.

We abbreviate $\tilde{M}_i = M_0 = (y_0, c_0), \ldots, M_{i-1} = (y_{i-1}, c_{i-1})$. The cumulative distribution function of the distance of one randomly selected ID is given by $F_{d,l}$. Then the probability distribution of the γ smallest values of k_d independent identically distributed random IDs is obtained as

$$\begin{aligned}
&P(X_1 = (\delta_1, ..., \delta_\gamma) | X_0 = d, L_d = l) \\
=&P(M_1 = (y_1, c_1), \ldots, M_{\gamma'} = (y_{\gamma'}, c_{\gamma'})) | X_0 = d, L_d = l) \\
=&\big(1 - P(X_1 = \emptyset | X_0 = d, L_d = l)\big) \qquad (17) \\
&\cdot \prod_{i=1}^{\gamma'} P\left(M_i = (y_i, c_i) | X_0 = d, L_d = l, X_1 \neq \emptyset, \tilde{M}_i\right)
\end{aligned}$$

It remains to determine each factor in Eq. 17. We first treat the case $i < \gamma'$, for which we have to determine the probability that (i) all $C_{i-1} = k_d - \sum_{j=1}^{i-1} c_j$ bucket entries with distance exceeding $y_{i-1} + 1$ are at distance at least y_i to the target, and (ii) there are *exactly* c_i such entries. Hence event (ii) conditioned on event (i) corresponds to the event that a binomially distributed random variable with C_{i-1} trials and success probability $p_i = \frac{F_{d,l}(y_i) - F_{d,l}(y_i - 1)}{1 - F_{d,l}(y_i - 1)}$ has exactly c_i successes. Note that the number of trials C_i and the denominator $1 - F_{d,l}(y_i - 1)$ result from conditioning on $M_1, \ldots, M_{i-1}$ and event (i), respectively. Then

$$\begin{aligned}
&P\left(M_i = (y_i, c_i) | X_0 = d, L_d = l, X_1 \neq \emptyset, \tilde{M}_i\right) \\
=&P\left(M_i(1) \geq y_i | X_0 = d, L_d = l, X_1 \neq \emptyset, \tilde{M}_i\right) \qquad (18) \\
&\cdot P\left(M_i = (y_i, c_i) | X_0 = d, L_d = l, X_1 \neq \emptyset, \tilde{M}_i, M_i(1) \geq y_i\right) \\
=&\left(\frac{1 - F_{d,l}(y_i - 1)}{1 - F_{d,l}(y_{i-1})}\right)^{C_{i-1}} \binom{C_{i-1}}{c_i} p_i^{c_i} (1 - p_i)^{C_i}.
\end{aligned}$$

The last step uses $C_i = C_{i-1} - c_i$.

For the γ'-th distinct value, the probability that there are *at least* $c_{\gamma'}$ equal values rather than exactly $c_{\gamma'}$ values is derived. There might be other contacts with the same distance in the bucket, which are not part of the chosen α contacts. Similarly, to Eq. 18, the last factor in Eq. 17 is

$$\begin{aligned}
&P\left(M_{\gamma'} = (y_{\gamma'}, c_{\gamma'}) | X_0 = d, L_d = l, X_1 \neq \emptyset, \tilde{M}_{\gamma'}\right) \qquad (19) \\
=&\left(\frac{1 - F_{d,l}(y_{\gamma'})}{1 - F_{d,l}(y_{\gamma'-1})}\right)^{C_{\gamma'-1}} \\
&\left(1 - \sum_{j=0}^{c_{\gamma'}-1} \binom{C_{\gamma'-1}}{j} p_{\gamma'}^j (1 - p_{\gamma'})^{C_{\gamma'-1}-j}\right).
\end{aligned}$$

Note that $1 - p_i = \frac{1 - F_{d,l}(y_i)}{1 - F_{d,l}(y_i - 1)}$ and hence

$$\begin{aligned}
&p_i^{c_i} (1 - p_i)^{C_i} \\
&= (F_{d,l}(y_i) - F_{d,l}(y_i - 1))^{c_i} \frac{(1 - F_{d,l}(y_i))^{C_i}}{(1 - F_{d,l}(y_i - 1))^{C_{i-1}}},
\end{aligned}$$

so that the claim follows by inserting Eqs. 18 and 19 in Eq. 17 and canceling. ∎

This completes our derivation of the hop distribution. However, our main goal is to provide an efficient computation. In the next section, we show that indeed the computation costs can be reduced to polylog complexity in the network size at a slight loss in accuracy.

V. Complexity Analysis

In the first part of this section, we determine the space and computation complexity of deriving the hop distribution. Finding that the complexity is at least $\mathcal{O}\left(b^{2\alpha}\right)$, we evaluate how a reduction of the common ID space size of $b = 128$ and $b = 160$ affects the accuracy.

A. Space complexity

We assume that the complete matrix T needs to be stored. In practice, the actual storage space can be slightly reduced by avoiding to store entries corresponding to impossible state transitions. However, from a state $s = (d_1, ..., d_\alpha)$ at least all states $(\delta_1, \ldots, \delta_\alpha)$ with $\delta_i < \delta_1$ for $i = 1 \ldots \alpha$ can be reached, so that the asymptotic complexity remains the same as for complete storage.

Lemma V.1. *The storage complexity for computing the hop distribution of a $\mathcal{K}(b, \alpha, \beta, k, L)$-system is $\mathcal{O}\left(\frac{1}{(\alpha!)^2} b^{2\alpha}\right)$.*

Proof: The storage complexity is dominated by the matrix $T \in \mathbb{R}^{|S|^2}$. Consequently, $|S|$ needs to be determined.

$$\begin{aligned}
|S| &= |\{\emptyset\} \cup \{s \in \mathbb{Z}_{b+1}^\alpha : s_j \leq s_{j+1}, j = 1 \ldots \alpha - 1\}| \\
&= 1 + \sum_{i_\alpha=0}^{b} \sum_{i_{\alpha-1}=0}^{i_\alpha} \cdots \sum_{i_1=0}^{i_2} 1 \\
&= \mathcal{O}\left(\int_0^b \int_0^{x_\alpha} \cdots \int_0^{x_2} 1 dx_1 dx_2 \ldots dx_\alpha\right) = \mathcal{O}\left(\frac{1}{\alpha!} b^\alpha\right).
\end{aligned}$$

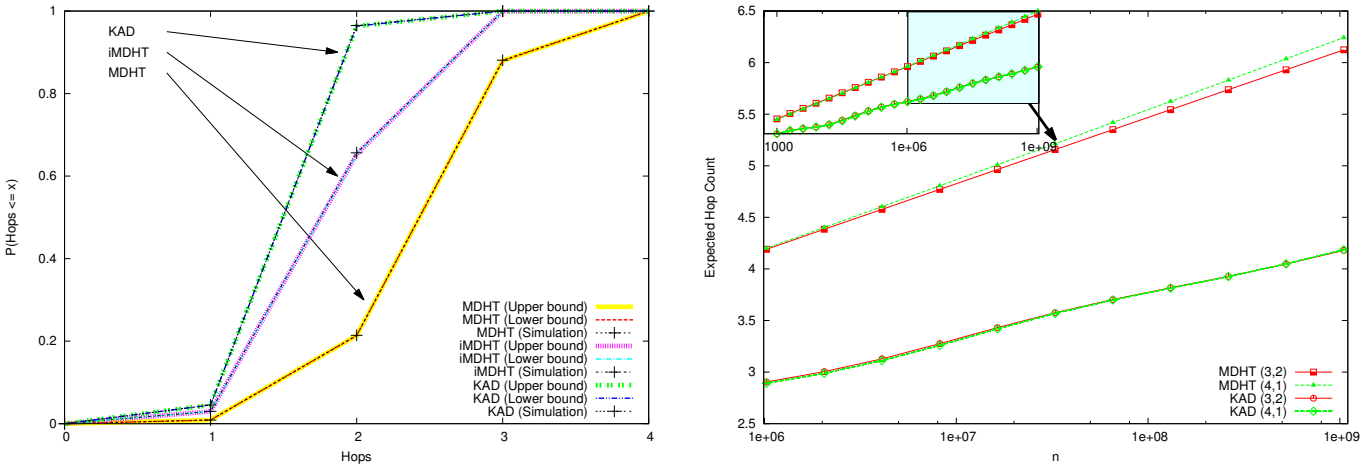
The size of the matrix T is S^2 and by this the space complexity is $\mathcal{O}\left(\frac{1}{(\alpha!)^2} b^{2\alpha}\right)$ as claimed. ∎

B. Computation complexity

We bound the computation complexity of the transition matrix. The bound holds for both T^{low} and T^{up}.

Lemma V.2. *The computation complexity is linear with regard to the network order n, and polynomial with regard to the ID space size b. More precisely, the number of basic operations is of order $\mathcal{O}\left(nb^{\alpha(\beta+1)}\right)$.*

The proof is similar to the one of Lemma V.1. We subsequently derive upper bounds on the computation costs considered in Theorem IV.1 and IV.2, Lemma IV.3 and IV.4, and the costs of the final matrix multiplication (Eq. 3). The individual steps are straightforward but rather lengthy, so that we defer them to our technical report [15].



(a) Validation

(b) Routing Parameter Study

Fig. 1: Exemplary model results: a) hop distributions for $10K$ nodes in comparison to simulation, and b) hop count for up to one billion nodes for MDHT and KAD with routing parameter $(\alpha, \beta) \in \{(3,2),(4,1)\}$ and stale entry rates $f \in \{0.0, 0.2\}$

C. Reducing the ID space size

From Lemma V.1 and Lemma V.2, we can see that both the storage and the computation complexity are polynomial in the bit-size b for a polynomial of high degree. In contrast, the dependence on the network order n is only linear for the computation complexity, whereas the storage complexity is independent of n. Though the dependence on α and β is exponential, both are usually small, less or equal than 4 in all existing or proposed systems. For instance, if $\alpha = 3$, the number of entries in the matrix T can be precisely computed as

$$
\begin{aligned}
&|\{F\} \cup \{(s_1,s_2,s_3) \in \mathbb{Z}_{b+1}^3 : s_1 \le s_2 \le s_3\}|^2 \\
&= \left(1 + \sum_{i_3=0}^{b} \sum_{i_2=0}^{i_3} \sum_{i_1=0}^{i_2} 1\right)^2 \\
&= \left(1 + \sum_{i_3=1}^{b+1} \frac{i_3(i_3+1)}{2}\right)^2 \\
&= \left(1 + 0.5 \cdot \sum_{i_3=1}^{b+1} (i_3^2 + i_3)\right)^2 \\
&= \left(1 + 0.5 \cdot \left(\frac{(b+1)(b+2)(2b+3)}{6} + \frac{(b+1)(b+2)}{2}\right)\right)^2 \\
&= \left(1 + \frac{(b+1)(b+2)(2b+6)}{12}\right)^2.
\end{aligned}
$$

In practice, $b = 128$ and $b = 160$ are typically used, corresponding to the length of MD-5 and SHA-1 hashes. Assuming 32 bit float numbers, the matrix T^* then needs roughly 500 GB and 1870 GB of storage.

Consequently, the computations are too expensive to present an alternative to extensive simulations. However, the accuracy can be expected to only depend exiguously on b, at least if the number of IDs is decisively higher than the number of nodes. Lemma V.3 provides an upper bound on the influence of b.

Lemma V.3. *Consider two Kademlia-type systems $K = \mathcal{K}(b, \alpha, \beta, k, L)$ and $\tilde{K} = \mathcal{K}(\tilde{b}, \alpha, \beta, \tilde{k}, \tilde{L})$, such that*

- $\tilde{b} < b$
- *$k[0..\tilde{b}] = \tilde{k}$, i.e. the vector $\tilde{k}$ contains exactly the $\tilde{b}+1$ first entries of k*
- *$L[0..\tilde{b}][0..\tilde{b}] = \tilde{L}$, i.e. $\tilde{L}$ is the upper left $\tilde{b}+1 \times \tilde{b}+1$ submatrix of L.*

The fractions of terminated queries $P^*(i)$ and $\tilde{P}^*(i)$ after i hops in K and $\tilde{K}$, respectively, differ by at most

$$|P^*(i) - \tilde{P}^*(i)| \le 1 - \sum_{j=0}^{\kappa} \binom{n}{j} p^j (1-p)^{n-j} \qquad (20)$$

with $p = 2^{-\tilde{b}} - 2^{-b}$, $\kappa = \min\{k_d : d = 0 \dots b\}$, and $* \in \{up, low\}$.

Proof: Note that the routing processes $(X_i)_{i \in \mathbb{N}_0}$ and $\left(\tilde{X}_i\right)_{i \in \mathbb{N}_0}$ are only different if there are at least κ nodes that share a common prefix of length at least $\tilde{b}$ with the target t, but not with a common prefix length of b. Recall that the query is considered successful in the next step after reaching a node with common prefix length $\tilde{b}$ by Assumption 6 in $\tilde{K}$, but not necessarily in K (see Section IV-A). The probability that two nodes share a common prefix of length $\tilde{b}$ to $b-1$ is $p = 2^{-\tilde{b}} - 2^{-b}$. The number of nodes with this property is hence $B(n,p)$ distributed. The claim follows. ∎

Based on Eq. 20, we now consider the trade-off between accuracy and computation cost in terms of the network order n.

Theorem V.4. *Consider two Kademlia-type systems K and $\tilde{K}$ as in Lemma V.3. For any $\epsilon > 0$, the error is $|P^*(i) - \tilde{P}^*(i)| \le \epsilon$ if $\tilde{b} \ge \lceil \log_2 \left(n \frac{2}{\ln 1/\epsilon}\right) \rceil$.*

Consequently, the storage complexity is $\mathcal{O}\left(log^{2\alpha} n\right)$ and the computation complexity is $\mathcal{O}\left(npolylog(n)\right)$ for a constant arbitrary small error ϵ.

Proof: For $\kappa = 0$ in Eq. 20, we can determine an upper bound on the minimal value for $\tilde{b}$ to achieve an error of less than ϵ. Set $C = \frac{2}{\ln 1/\epsilon}$ in the following. From $\epsilon \le 1 - (1-p)^n < 1 - \left(1 - 2^{-\tilde{b}}\right)^n$, it follows that $\tilde{b} \ge \log_2 \frac{1}{1-\epsilon^{1/n}}$. It

remains to show that for n large enough, $\frac{1}{1-\epsilon^{1/n}} < Cn$, which is equivalent to $\epsilon < (1-\frac{1}{Cn})^n$. Because $(1-\frac{1}{Cn})^n$ converges to $e^{-1/C}$, there exists n, such that

$$\left(1-\frac{1}{Cn}\right)^n > e^{-2/C} = e^{-ln1/\epsilon} = \epsilon.$$

Therefore, for n large enough, $\tilde{b} \geq \log_2\left(n\frac{2}{\ln 1/\epsilon}\right)$ ensures that $|P^*(i) - \tilde{P}^*(i)| \leq \epsilon$.

The storage complexity is $\mathcal{O}\left(log^{2\alpha}n\right)$ by Lemma V.1 with $\tilde{b} = \mathcal{O}(\log n)$, the computation complexity of $\mathcal{O}\left(npolylog(n)\right)$ follows from Lemma V.2. ∎

Note that by using Eq. 20 rather than the approximation in Theorem V.4, the bound on $\tilde{b}$ can be further reduced. However, Theorem V.4 proves that the number of bits needed for a certain accuracy grows logarithmically in the network size.

VI. Results

In this section, we show that our derivation closely approximates results from simulations. Furthermore, we evaluate the scalability of our approach, showing that networks of up to one billion nodes can be analyzed. In addition, we combine the scalability analysis with an exemplary study of the routing parameters α and β. Throughout the section, the number of bits b in the model was chosen such that a maximal error of $\epsilon = 0.001$ is guaranteed.

A. Simulations

We compare the results produced by our model with simulations. For this purpose, we implemented the three routing tables structures of MDHT, iMDHT, and KAD in OverSim [16], an event-based simulator for P2P overlays. 128-bit IDs were assigned uniformly at random. The maintenance protocols were implemented as specified by the KAD implementation in eMule. The statistics were gathered after an initial stabilization phase of 2,000 simulation seconds, the actual measurement time was chosen to be 12,500 simulation seconds. Every online node queried for a random destination node every 300 simulation seconds. The source code for the simulation study is available upon request. The routing algorithm was parametrized by $\alpha = 3$ and $\beta = 2$. A network size of $10K$ was chosen. The simulation results were averaged over 10 runs.

Figure 1a presents the upper and lower bounds on the hop distribution in comparison to the simulation results. The upper and the lower bound are separated by less than $\epsilon = 0.001$, and simulations and model agree within a 95% confidence interval for all data points.

B. Scalability and Exemplary Parameter Study

In order to show that our model scales well up to the millions or even billions of nodes, we computed the hop distribution for both MDHT and KAD for networks of size $2^i \cdot 1000$ for $i = 1 \ldots 20$ (so up to more than 1 billion). Besides verifying the scalability, we also compared the standard routing algorithm parameters $(\alpha, \beta) = (3, 2)$ to $(\alpha, \beta) = (4, 1)$, which are used in KAD and the BitTorrent client uTorrent in MDHT, respectively. Figure 1b displays the expected hop. Changing the routing parameters to $(4, 1)$ only improves the performance up to a network size of about $400K$ for MDHT, but achieved a slightly lower expected hop count for KAD for all considered network sizes. Remark that KAD uses $(\alpha, \beta) = (3, 2)$, while the use of $(\alpha, \beta) = (4, 1)$ in MDHT increases both the hop count and the overhead per step (contacting 4 rather than 3 nodes) for typical network sizes of several millions of nodes. The advantage of returning $\beta > 1$ is more pronounced under churn, because it mitigates the impact of non-responding nodes. These results show the advantage of our approach over simulations: Since network simulators only scale to a few ten thousands of nodes, a simulation study might suggest the change in parameters, but the observed benefit achieved by a higher parallelism does not transfer to larger networks.

VII. Conclusion

We have introduced a scalable accurate characterization of the hop count distribution in Kademlia-type systems. Our solution is the first which enables to compute the hop distribution for a wide range of parameters. In particular, we manage to integrate parallelism into our model, in contrast to previous solutions. Furthermore, we demonstrated the utility of our model by analyzing common design decisions in Kademlia-type systems, showing that returning more than one contact per query is essential for achieving shorter routes both in static and in dynamic environments. Questions regarding the routing table structure, in particular the effect of maintaining multiple buckets per level, as well as optimizing Kademlia routing to given environments and system constraints, remain future work.

References

[1] Liang Wang and Jussi Kangasharju. Measuring large-scale distributed systems: case of bittorrent mainline dht. In *P2P*, 2013.
[2] Moritz Steiner et al. A Global View of KAD. In *IMC*, 2007.
[3] Daniel Stutzbach and Reza Rejaie. Improving Lookup Performance Over a Widely-Deployed DHT. In *INFOCOM*, 2006.
[4] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *IPTPS*, 2002.
[5] Raul Jimenez et al. Sub-Second Lookups on a Large-Scale Kademlia-Based Overlay. In *P2P*, 2011.
[6] Peng Wang et al. Attacking the KAD Network. In *SecureComm*, 2008.
[7] Moritz Steiner et al. Evaluating and improving the content access in KAD. *Peer-to-Peer Networking and Applications*, 2010.
[8] Jarret Falkner et al. Profiling a Million User DHT. In *IMC*, 2007.
[9] Scott Crosby and Dan Wallach. An Analysis of BitTorrent's Two Kademlia-Based DHTs. Technical report, Rice University, 2007.
[10] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *STOC*, 2000.
[11] Gurmeet Singh Manku et al. Know thy neighbor's neighbor: the power of lookahead in randomized p2p networks. In *STOC*, 2004.
[12] Pierre Fraigniaud and George Giakkoupis. The effect of power-law degrees on the navigability of small worlds. In *PODC*, 2009.
[13] A. Spognardi and R. Di Pietro. A formal framework for the performance analysis of p2p networks protocols. In *IPDPS*, 2006.
[14] Idris Rai et al. Performance modelling of peer-to-peer routing. In *IPDPS*, 2007.
[15] Stefanie Roos et al. Comprehending Kademlia Routing - A Theoretical Framework for the Hop Count Distribution. Technical report, arXiv CoRR, 2013.
[16] Ingmar Baumgart et al. Oversim: A scalable and flexible overlay framework for simulation and real network applications. In *P2P*, 2009.