The Analysis of Kademlia for random IDs

Xing Shi Cai   Luc Devroye*

School of Computer Science, McGill University of Montreal, Canada,

xingshi.cai@mail.mcgill.ca   lucdevroye@gmail.com

May 12, 2015

Abstract

Kademlia [7] is the de facto standard searching algorithm for P2P (peer-to-peer) networks on the Internet. In our earlier work [2], we introduced two slightly different models for Kademlia and studied how many steps it takes to search for a target node by using Kademlia's searching algorithm. The first model, in which nodes of the network are labeled with deterministic IDs, had been discussed in that paper. The second one, in which nodes are labeled with random IDs, which we call the Random ID Model, was only briefly mentioned. Refined results with detailed proofs for this model are given in this paper. Our analysis shows that with high probability it takes about $c \log n$ steps to locate any node, where n is the total number of nodes in the network and c is a constant that does not depend on n.

1 Introduction to Kademlia

A P2P (peer-to-peer) network [11] is a decentralized computer network which allows participating computers (*nodes*) to share resources. Some P2P networks have millions of live nodes. To allow searching for a particular node without introducing bottlenecks in the network, a group of algorithms called DHT (Distributed Hash Table) [1] was invented in the early 2000s, including Plaxton's algorithm [8], Pastry [10], CAN [9], Chord [13], Koorde [6], Tapestry [15], and Kademlia [7]. Among them, Kademlia is most widely used in today's Internet.

In Kademlia, each node is assigned an ID selected uniformly at random from $\{0, 1\}^d$ (ID *space*), where d is usually 128 [12] or 160 [3]. The *distance* between two nodes is calculated by performing the bitwise exclusive or (XOR) operation over their IDs and taking the result as a binary number. (In this work *distance* and *closeness* always refer to the XOR distance between IDs.)

Roughly speaking, a Kademlia node keeps a table of a few other nodes (*neighbors*) whose distances are sufficiently diverse. So when a node searches for an ID, it always has some neighbors close to its target. By inquiring these neighbors, and these neighbors' neighbors, and so on, the node that is closest to the target ID in the network will be found eventually. Other DHTs work in similar ways. The differences mainly come from how distance is defined and how neighbors are chosen. For a more detailed survey of DHTs, see [1].

---

2   The Random ID Model

This section briefly reviews the Random ID Model for Kademlia defined in [2]. Let $d \geq \log_2 n$ be the length of n binary IDs $X_1, \ldots, X_n$ chosen uniformly at random from $\{0,1\}^d$ without replacement. Consider n nodes indexed by $i \in \{1, \ldots, n\}$. Let X_i be the ID of node i.

Given two IDs $x = (x_1, \ldots, x_d), y = (y_1, \ldots, y_d)$, their XOR distance is defined by

$$\delta(x,y) = \sum_{j=1}^{d} (x_j \oplus y_j) \times 2^{d-j}.$$

where $\oplus$ is the XOR operator

$$u \oplus v = \begin{cases} 1 & \text{if } u \neq v, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\ell(x,y)$ be the length of the common prefix of x and y. The n nodes can be partitioned into $d+1$ parts by their common prefix length with x via

$$\mathcal{S}(x,j) = \{i : 1 \leq i \leq n, \ \ell(x, X_i) = j\}, \qquad 0 \leq j \leq d.$$

For each $1 \leq i \leq n$, d tables (*buckets*) of size at most k are kept, where k is a fixed positive integer. Buckets are indexed by $j \in \{0, \ldots, d-1\}$. The bucket j is filled with $\min\{k, |\mathcal{S}(X_i, j)|\}$ indices drawn uniformly at random from $\mathcal{S}(X_i, j)$ without replacement. Note that the first j bits of X_s, if $s \in \mathcal{S}(X_i, j)$, agree with the first j bits of X_i, but the $(j+1)$-th bit is different.

Searching for $y \in \{0,1\}^d$ initiated at node i proceeds as follows. Given that $\ell(y, X_i) = j$, y can only be in $\mathcal{S}(X_i, j)$. Thus, all indices from the bucket j of i are retrieved, say $i_1, \ldots, i_k$. From them, the one having shortest distance to y is selected as i^*. (In fact, any selection algorithm would be sufficient for the results of this paper.) Note that

$$\ell(y, X_{i^*}) = \max_{1 \leq r \leq k} \ell(y, X_{i_r}).$$

Thus the choice of i^* does not depend on the exact distances from $X_{i_1}, \ldots, X_{i_k}$ to y. Therefore, instead of the XOR distance, only the length of common prefix is needed in the following analysis of searching.

The search halts if $y = X_i$ or if the bucket is empty. In the latter case, X_i is closest to y among all nodes. Otherwise we continue from i^*. Since $\ell(y, X_{i^*}) > \ell(y, X_i)$, the maximal number of steps before halting is bounded by d. Let T_i be the number of steps before halting in the search of y when started from i (*searching time*). Then $T_i = T_{i^*} + 1$.

Treating $X_1, \ldots, X_n$ as strings consisting of zeros and ones, they can be represented by a tree data structure called *trie* [14]. The $\mathcal{S}(x,j)$'s can be viewed as subtrees. Filling buckets is equivalent to choosing at most k leaves from each of these subtrees. Fig. 1 gives an example of an ID trie.

3   Main Results

The structure of the model is such that nothing changes if $X_1, \ldots, X_n, y$ are replaced by their coordinate-wise XOR with a given vector $z \in \{0,1\}^d$. This is a mere rotation of the hypercube. Thus, it can be assumed without loss of generality that $y = (1, 1, \ldots, 1)$, the rightmost branch in the ID trie.

2

Figure 1: An example of Kademlia ID trie. Given an ID $x = (1, 0, 0)$, the trie is partitioned into subtrees $\mathcal{S}(x, 0), \mathcal{S}(x, 1)$ and $\mathcal{S}(x, 2)$. Node x maintains a bucket for each of these subtrees containing at most k nodes from the corresponding subtree.

If $d \sim c \log_2 n$ for some $c \geq 1$, the searching time is $O(\log n)$, which is undoubtedly a contributing factor in Kademlia's success. If $d = \omega(n)$, then it is not a useful upper bound of searching time any more. However, in some probabilistic sense, T_i can be much smaller than $\log_2 n$—it can be controlled by the parameter k, which measures the amount of storage consumed by each node. The aim of this work is to investigate finer properties of these random variables. In particular, the following theorem is proved:

Theorem 1. *Assume that $d \geq \log_2 n$. Let $k > 0$ be a fixed integer. Let $\xrightarrow{p}$ denote convergence in probability. Then*

$$\frac{T_1}{\log_2 n} \xrightarrow{p} \frac{1}{\mu_k}, \qquad as\ n \to \infty,$$

$$\frac{\boldsymbol{E}T_1}{\log_2 n} \to \frac{1}{\mu_k}, \qquad as\ n \to \infty,$$

where μ_k is a function of k only:

$$\mu_k = \sum_{j=1}^{\infty} 1 - \left(1 - \frac{1}{2^{j-1}}\right)^k.$$

In particular, $\mu_1 = 2$.

In the rest of the paper, we first show that once the search reaches a node that shares a common prefix of length about $\log n$ with y, the search halts in $o(\log n)$ steps. Thus it suffices to prove Theorem 1 for the time that it takes for this event to happen. Then we show that the ID trie is well balanced with high probability. Thus when n is a power of 2, we can couple the search in the original trie with a search in a trie that is a complete binary tree. It proves the theorem for this special case. After that, we give a sketch of how to deal with general n. At the end we briefly summarize some implications of the theorem.

3

4   The Tail of the Search Time

To keep the notation simple, let $m = \log_2 n$ and and note that m is not necessarily integer-valued. Also, for analytic purposes, define

$$J = \min\left\{j : \frac{n}{2^{j+1}} \leq m^4\right\}.$$

Since $n/2^J > m^4$ and $n/2^{J+1} \leq m^4$,

$$J < \log_2 \frac{n}{m^4} = m - 4\log_2 m \leq m, \tag{1}$$

$$J \geq \log_2 \frac{n}{m^4} - 1 = m - 4\log_2 m - 1. \tag{2}$$

The importance of J follows from the fact that once the search reaches a node i with $\ell(X_i, y) \geq J$, it takes very few steps to finish. Let T_1' be the number of search steps that depart from a node in the set $\mathcal{S}(y, j)$ for some $j < J$, with the very first node in the search being 1.

Lemma 1. *Theorem 1 follows if*

$$\frac{T_1'}{\log_2 n} \xrightarrow{p} \frac{1}{\mu_k}, \qquad \text{as } n \to \infty.$$

Proof. Let $T_1'' = T_1 - T_1'$. T_1'' counts steps of the search departing from a node in $\bigcup_{j=J}^{d-1} \mathcal{S}(y, j)$. Thus

$$T_1'' \leq \sum_{j \geq J}^{d-1} \mathbf{1}_{[|\mathcal{S}(y,j)|>0]}.$$

Noting that

$$\mathbf{E}|\mathcal{S}(y, j)| = \frac{n}{2^{j+1}}, \tag{3}$$

by linearity of expectation,

$$\mathbf{E}T_1'' \leq \sum_{j \geq J}^{d-1} \mathbf{P}\left\{|\mathcal{S}(y,j)| \geq 1\right\} \leq \sum_{j \geq J}^{d-1} \min\{\mathbf{E}|\mathcal{S}(y,j)|, 1\}$$

$$\leq \sum_{j \geq J}^{d-1} \min\left\{\frac{n}{2^{j+1}}, 1\right\} \qquad \text{(by (3))}$$

$$\leq \sum_{j \geq J}^{d-1} \mathbf{1}_{[2^{j+1}<n]} + \sum_{j \geq J}^{d-1} \mathbf{1}_{[2^{j+1} \geq n]} \times \frac{n}{2^{j+1}}$$

$$\leq 4\log_2 \log_2 n + 2 \qquad \text{(by (2)).}$$

Thus, for all $\epsilon > 0$ fixed,

$$\mathbf{P}\left\{T_1'' \geq \epsilon \log_2 n\right\} \leq \frac{\mathbf{E}T_1''}{\epsilon \log_2 n} = o(1),$$

4

Therefore $T_1''/\log_2 n \xrightarrow{p} 0$. For the expectation, note that

$$\frac{\mathbf{E} T_1'}{\log_2 n} \to \frac{1}{\mu_k}, \qquad \text{as } n \to \infty,$$

by the lemma's assumption and the fact that $T_1'/\log_2 n \le 1 < \infty$. □

5 Good Tries and Bad Tries

Since the tail of search does not matter, define a new partition S_j of all nodes by merging subtrees $\mathcal{S}(y, j)$ for $j \ge J$ as follows:

$$S_j = \begin{cases} \mathcal{S}(y, j) & \text{if } 0 \le j < J, \\ \bigcup_{i=J}^{d} \mathcal{S}(y, i) & \text{if } j = J. \end{cases}$$

Let $N_j = |S_j|$. It follows from (3) that

$$\mathbf{E} N_j = \begin{cases} n/2^{j+1} & \text{if } 0 \le j < J, \\ n/2^J & \text{if } j = J, \end{cases} \tag{4}$$

or simply $\mathbf{E} N_j = n/2^{(j+1) \wedge J}$, where $a \wedge b \overset{\text{def}}{=} \min\{a, b\}$. Note that N_j is hypergeometric with parameters

$$\left(n, \frac{2^d}{2^{(j+1) \wedge J}}, 2^d - \frac{2^d}{2^{(j+1) \wedge J}} \right),$$

i.e., it corresponds to the selection of n balls without replacement from an urn of 2^d balls of which $2^d/2^{(j+1) \wedge J}$ are white [5, chap. 6.3].

The analysis of T_1' can be simplified if the N_j's are all close to their expectations. To be precise, let $\alpha = m^{-3/2}$ be the *accuracy parameter*. An ID trie is *good*, if

$$|N_j - \mathbf{E} N_j| \le \alpha \times \mathbf{E} N_j,$$

for all $0 \le j \le J$. Otherwise it is called *bad*.

Lemma 2. *The probability that an* ID *trie is bad is* $o(1)$.

Proof. It follows from the union bound that

$$\mathbf{P}\left\{ \bigcup_{j=0}^{J} [|N_j - \mathbf{E} N_j| > \alpha \times \mathbf{E} N_j] \right\} \le \sum_{j=0}^{J} \mathbf{P}\left\{ [|N_j - \mathbf{E} N_j| > \alpha \times \mathbf{E} N_j] \right\}$$

$$\le \sum_{j=0}^{J} \frac{\mathbf{Var}(N_j)}{(\alpha \times \mathbf{E} N_j)^2} \qquad \text{(by Chebyshev's inequality)}$$

$$\le \sum_{j=0}^{J} \frac{\mathbf{E} N_j}{(\alpha \times \mathbf{E} N_j)^2} \qquad (N_j \text{ is hypergeometric})$$

$$\le \frac{1}{\alpha^2} \times \sum_{j=0}^{J} \frac{2^{j+1}}{n} \qquad \text{(by (4))}$$

$$\le m^3 \times \frac{2^{J+2}}{n} = o(1). \qquad \left(\text{since } \frac{n}{2^J} > m^4 \right)$$
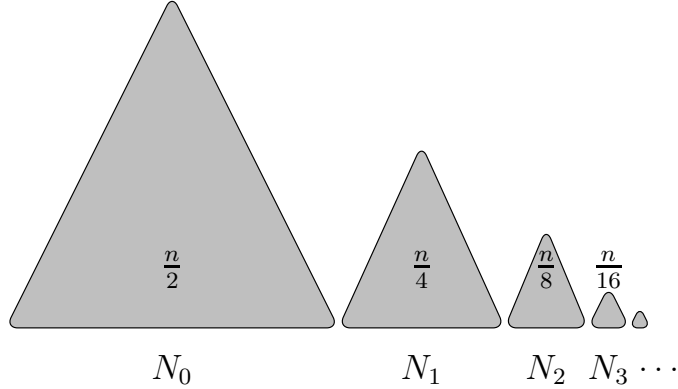
5

Figure 2: The approximate sizes of subtrees in a good trie.

The fact used here is that $\mathbf{Var}(N_j) \leq \mathbf{Var}(N_j')$ where N_j' is binomial $(n, 1/2^{(j+1)\wedge J})$. For the binomial, $\mathbf{Var}(N_j') \leq \mathbf{E}N_j' = \mathbf{E}N_j$. $\qquad\square$

6 Proof when n Is a Power of 2

In this section, n is assumed to be a power of 2, i.e., m is an integer. The general case is treated in the next section.

6.1 A Perfect Trie

Construct a coupled ID trie consisting of $Y_1, \ldots, Y_n$ as follows. If $N_j \geq \mathbf{E}N_j$, i.e., the size of the subtree $\mathcal{S}_j$ is at least its expectation, let $Y_i = X_i$ for the $\mathbf{E}N_j$ smallest indices in $\mathcal{S}_j$. After this preliminary coupling, some Y_i's are undefined. The indices i for which Y_i are undefined go into a global pool $\mathcal{G}$ of size

$$\sum_{j=0}^{J} \max\{N_j - \mathbf{E}N_j, 0\}.$$

For a good trie, the size of the pool is at most

$$\sum_{j=0}^{J} \alpha \times \mathbf{E}N_j = \alpha \times \mathbf{E}\sum_{j=0}^{J} N_j = \alpha n.$$

For a subtree $\mathcal{S}_j$ of size $N_j < \mathbf{E}N_j$, take $\mathbf{E}[N_j] - N_j$ indices i from $\mathcal{G}$ and assign Y_i a value, that is different from all other Y_s's, and that has $\ell(Y_i, y) \wedge J = j$. Subtrees of this new trie have fixed sizes of

$$|\{i : \ell(Y_i, y) \wedge J = j\}| = \mathbf{E}N_j = \frac{n}{2^{(j+1)\wedge J}}, \qquad 0 \leq j \leq J. \tag{5}$$

A trie like this is called *perfect*. Indices i for which $X_i \neq Y_i$, i.e., $i \in \mathcal{G}$, are called *ghosts*. Other indices are called *normal*.

Next, refill the buckets according to the perfect trie, but keep buckets of normal indices containing no ghosts unchanged. Observe that a search step departing at a normal index i

6

proceeds precisely the same in both tries if bucket j (with $j = \ell(Y_i, y)$) of i does not contain ghosts. Assuming that the original trie is good, the probability that a bucket that corresponds to $\mathcal{S}_j$ for some $j \leq J$ contains a ghost is not more than $k\alpha$. This is because in the newly constructed prefect trie, the subtree $\mathcal{S}_j$ contains no more than α proportion of ghost nodes.

Let T_1^* denote the number of search steps starting from node 1 via node i with $\ell(Y_i, y) < J$ in the perfect trie. Then $[T_1^* \neq T_1'] \subseteq B$, where B is the event that at least one node in the buckets encountered during a search is a ghost. Let A be the event that the trie is good. It follows from Lemma 2 that

$$\mathbf{P}\{T_1^* \neq T_1'\} \leq \mathbf{P}\{B\} \leq \mathbf{P}\{B, A\} + \mathbf{P}\{A^c\} \leq J \times k\alpha + o(1) = o(1).$$

Therefore, Theorem 1 follows if

$$\frac{T_1^*}{\log_2 n} \xrightarrow{p} \frac{1}{\mu_k}, \qquad \text{as } n \to \infty.$$

6.2 Filling the Buckets with Replacement

To deal with the problem that buckets are filled by sampling without replacement, another coupling argument is needed. Let p_j be the probability that the k items sampled with replacement from a set of size $n/2^{j+1}$ are not all distinctive. Observe that by the union bound,

$$p_j \leq \binom{k}{2} \frac{2^{j+1}}{n} \leq \frac{k^2 2^j}{n}.$$

If $\ell(Y_i, y) = j < J$, then bucket j of i has k elements drawn without replacement from

$$\mathcal{S} = \{s : \ell(Y_s, y) \geq j + 1\}, \quad 0 \leq j < J.$$

Observe that

$$|\mathcal{S}| = \frac{n}{2^{j+2}} + \frac{n}{2^{j+3}} + \cdots + \frac{n}{2^J} + \frac{n}{2^J} = \frac{n}{2^{j+1}}.$$

Hence, with probability $1 - p_j$, the sampling can be seen as having been carried out with replacement.

The coupling is as follows: for all i with $\ell(Y_i, y) = j$ and all $0 \leq j < J$, mark bucket j of i with probability p_j. When a bucket is marked, replace its entries with k new entires drawn with replacement conditioned on the existence of at least one duplicate entry. In this way, all bucket entries are for a sampling with replacement. Let the search time, starting still from 1, be denoted by T_1^{**}. Let D be the event that during the search a marked bucket is encountered. Observe that $[T_1^* \neq T_1^{**}] \subseteq D$. Therefore

$$\mathbf{P}\{T_1^* \neq T_1^{**}\} \leq \mathbf{P}\{D\} \leq \sum_{j=0}^{J-1} p_j \leq \sum_{j=0}^{J-1} \frac{k^2 2^j}{n} < \frac{k^2 2^J}{n} < \frac{2k^2}{m^4} = o(1).$$

So Theorem 1 follows if

$$\frac{T_1^{**}}{\log_2 n} \xrightarrow{p} \frac{1}{\mu_k}, \qquad \text{as } n \to \infty.$$

6.3 Analyzing T_1^{**} Using a Sum of I.I.D. Random Variables

Let $\Delta_0 = \ell(Y_1, y)$. Assume that step t of the search departs from node i and reaches node i^*. Let $\Delta_t = \ell(Y_{i^*}, y) - \ell(Y_i, y)$, i.e., Δ_t represents the progress in this step. Then

$$T_1^{**} = \inf \left\{ t : \sum_{s=0}^{t} \Delta_s \geq J \right\}.$$

Due to the recursive structure of a perfect trie, $\Delta_1, \Delta_2, \ldots$, although not i.i.d., should have very similar distributions. This intuition leads to the following analysis of T_1^{**} by studying a sum of i.i.d. random variables.

One observation allows us to deal with truncated version of Δ_t's is as follows:

Lemma 3. *Let $w_0, w_1, \ldots$ be a sequence of real numbers with $\sum_{t \geq 0} w_t = \infty$. Define*

$$\overline{w}_t = w_t \wedge \left(M - \sum_{s=0}^{t-1} \overline{w}_s \right), \qquad t = 0, 1, 2 \ldots,$$

where M is also a real number. Then

$$\inf \left\{ t : \sum_{s=0}^{t} w_s \geq M \right\} = \inf \left\{ t : \sum_{s=0}^{t} \overline{w}_s \geq M \right\},$$

where we define the infimum of an empty set to be ∞.

Proof. Let $\tau = \inf \left\{ t : \sum_{s=0}^{t} w_s \geq M \right\}$. If $\tau = \infty$ or $\tau = 0$, the lemma is trivially true. So we assume $0 < \tau < \infty$. By induction on t, one can show that $\overline{w}_t = w_t$ if $t < \tau$. Since $0 < \tau$, we have $\overline{w}_0 = w_0$, which is the induction basis. If $\overline{w}_s = w_s$ for all $0 \leq s \leq t-1$ and $t < \tau$, then

$$\overline{w}_t = w_t \wedge \left(M - \sum_{s=0}^{t-1} \overline{w}_s \right) = w_t \wedge \left(M - \sum_{s=0}^{t-1} w_s \right) = w_t.$$

Therefore $\sum_{s=0}^{t} w_s < M$ if and only if $\sum_{s=0}^{t} \overline{w}_s < M$. □

Let $\overline{\Delta}_t = \Delta_t \wedge \left(J - \sum_{s=0}^{t-1} \overline{\Delta}_s \right)$. It follows from the previous lemma that

$$T_1^{**} = \inf \left\{ t : \sum_{s=0}^{t} \overline{\Delta}_s \geq J \right\},$$

which is quite convenient as the distribution of $\overline{\Delta}_t$ is easy to compute.

Assume again that step t of the search departs from node i with $\ell(Y_i, y) = j < J$. Consider one item, say z, in bucket j of i. Recall that z is selected uniformly at random from all indices r with $\ell(r, y) \geq j + 1$. Thus it follows from the structure of a perfect trie, which is given by (5), that

$$\mathbf{P}\left\{ \ell(Y_z, y) = s \right\} = \frac{\frac{n}{2^{s+1}}}{\frac{n}{2^{j+2}} + \frac{n}{2^{j+3}} + \cdots + \frac{n}{2^J} + \frac{n}{2^J}} = \frac{1}{2^{s-j}}, \qquad j+1 \leq s < J,$$

$$\mathbf{P}\left\{ \ell(Y_z, y) \geq J \right\} = \frac{\frac{n}{2^J}}{\frac{n}{2^{j+2}} + \frac{n}{2^{j+3}} + \cdots + \frac{n}{2^J} + \frac{n}{2^J}} = \frac{1}{2^{J-j-1}}.$$

Or shifted by $-j$,

$$\mathbf{P}\{\ell(Y_z, y) - j = s\} = \frac{1}{2^s}, \qquad 1 \le s < J - j,$$

$$\mathbf{P}\{\ell(Y_z, y) - j \ge J - j\} = \frac{1}{2^{J-j-1}}.$$

If truncated by $J - j$, we obtain

$$\mathbf{P}\{(\ell(Y_z, y) - j) \wedge (J - j) = s\} = \frac{1}{2^{s \wedge (J-j-1)}}, \qquad 1 \le s \le J - j.$$

Note that this is *exactly* the distribution of a geometric $(1/2)$ truncated by $J - j$.

Recall that among all the values of $\ell(\,\cdot\,, y)$ given by items in the bucket j of i, the one chosen as the next stop of the search gives the maximum. Thus

$$\Delta_t = \max_{z \in \text{bucket } j} \{\ell(Y_z, y) - j\}.$$

Let $Z_1, Z_2, \ldots$ be i.i.d. geometric $(1/2)$. Let $V = \max\{Z_1, \ldots, Z_k\}$. Then

$$
\begin{aligned}
\overline{\Delta}_t &= \Delta_t \wedge (J - j) \\
&= \max_{z \in \text{bucket } j} \{(\ell(Y_z, y) - j)\} \wedge (J - j) \\
&= \max_{z \in \text{bucket } j} \{(\ell(Y_z, y) - j) \wedge (J - j)\} \\
&\overset{\mathcal{L}}{=} \max\{Z_1 \wedge (J - j), \ldots, Z_k \wedge (J - j)\} \\
&= \max\{Z_1, \ldots, Z_k\} \wedge (J - j) \\
&= V \wedge (J - j).
\end{aligned}
$$

Let V_0 be a geometric $(1/2)$ minus one. Then $V_0 \wedge d \overset{\mathcal{L}}{=} \Delta_0$. Let $V_1, V_2, \ldots$ be i.i.d. random variables distributed as V. Let $\overline{V}_t = V_t \wedge (J - \sum_{s=0}^{t-1} \overline{V}_s)$. Using induction and the previous argument about $\overline{\Delta}_t$, one can show that

$$\sum_{s=0}^{t} \overline{V}_s \overset{\mathcal{L}}{=} \sum_{s=0}^{t} \overline{\Delta}_s \qquad t = 0, 1, \ldots. \tag{6}$$

For the induction basis, note that

$$\overline{\Delta}_0 = \Delta_0 \wedge J \overset{\mathcal{L}}{=} (V_0 \wedge d) \wedge J = V_0 \wedge J = \overline{V}_0.$$

Assume that $\sum_{s=0}^{t-1} \overline{V}_s \overset{\mathcal{L}}{=} \sum_{s=0}^{t-1} \overline{\Delta}_s$ for some $t > 0$. Then for all $0 \le i \le J$,

$$
\begin{aligned}
\mathbf{P}\left\{\sum_{s=0}^{t} \overline{\Delta}_s = i\right\} &= \sum_{j=0}^{i} \mathbf{P}\left\{\overline{\Delta}_t = i - j \,\bigg|\, \sum_{s=0}^{t-1} \overline{\Delta}_s = j\right\} \mathbf{P}\left\{\sum_{s=0}^{t-1} \overline{\Delta}_s = j\right\} \\
&= \sum_{j=0}^{i} \mathbf{P}\{V_t \wedge (J - j) = i - j\} \mathbf{P}\left\{\sum_{s=0}^{t-1} \overline{V}_s = j\right\} \\
&= \sum_{j=0}^{i} \mathbf{P}\left\{\left[\overline{V}_t = i - \sum_{s=0}^{t-1} \overline{V}_s\right] \cap \left[\sum_{s=0}^{t-1} \overline{V}_s = j\right]\right\} = \mathbf{P}\left\{\sum_{s=0}^{t} \overline{V}_s = i\right\}.
\end{aligned}
$$

Thus (6) is proved. It then follows from Lemma 3 and (6) that

$$T_1^{**} \overset{\mathcal{L}}{=} \inf \left\{ t : \sum_{s=0}^{t} \overline{V}_s \geq J \right\} = \inf \left\{ t : \sum_{s=0}^{t} V_s \geq J \right\},$$

which makes T_1^{**} much easier to analyze.

Since $V < s$ if and only if $Z_1, \ldots, Z_k$ are all smaller than s,

$$\mathbf{P}\{V < s\} = \prod_{r=1}^{k} \mathbf{P}\{Z_r < s\} = \left(1 - \frac{1}{2^{s-1}}\right)^k.$$

Therefore, by definition of μ_k,

$$\mathbf{E}V = \sum_{s=1}^{\infty} \mathbf{P}\{V \geq s\} = \sum_{s=1}^{\infty} 1 - \left(1 - \frac{1}{2^{s-1}}\right)^k = \mu_k.$$

Readers familiar with renewal theory [4, chap. 4.4] can immediately see that

$$\frac{T_1^{**}}{\log_2 n} = \frac{T_1^{**}}{J} \times \frac{J}{\log_2 n} \overset{p}{\to} \frac{1}{\mathbf{E}V} = \frac{1}{\mu_k},$$

which completes the proof of Theorem 1 for n which is power of 2. The following Lemma gives some more details.

Lemma 4. *If* $\tau = \inf \left\{ t : \sum_{s=0}^{t} V_s \geq M \right\}$,

$$\frac{\tau}{M/\mathbf{E}V} \overset{p}{\to} 1, \qquad as\ M \to \infty.$$

Proof. Since $V_0 + 1$ is geometric $(1/2)$,

$$\mathbf{P}\{V_0 + 1 \leq s\} = 1 - \frac{1}{2^s} \geq \left(1 - \frac{1}{2^s}\right)^k = \mathbf{P}\{V_1 \leq s\}.$$

In other words, $V_0 \preceq V_1$, where $\preceq$ denotes stochastical ordering. Let

$$\tau' = \inf \left\{ t : \sum_{s=1}^{t} V_s \geq M \right\}, \qquad \tau'' = \inf \left\{ t : \sum_{s=0}^{t} V_{s+1} \geq M \right\} = \tau' - 1.$$

Then $\tau'' \preceq \tau$ and $\tau \leq \tau'$. By the strong law of large numbers, both τ'/M and τ''/M converge to $1/\mathbf{E}V$ almost surely. Therefore $\tau/M \overset{p}{\to} 1/\mathbf{E}V$. $\qquad\square$

7 Proof for the General Case

In this section, the proof Theorem 1 for n an arbitrary integer is only sketched as most methods used here are very similar to those in the previous section.

7.1 An Almost Perfect Trie

When n is not power of 2, $\mathbf{E}N_j = n/2^{(j+1)\wedge J}$ is not guaranteed to be an integer. So a perfect trie is not well defined any more. However, let us define

$$b_j = \begin{cases} \lceil \mathbf{E}N_j \rceil = \lceil \frac{n}{2^{j+1}} \rceil & 0 \leq j < J, \\ n - \sum_{s=0}^{J-1} b_s = n - \sum_{s=0}^{J-1} \lceil \frac{n}{2^{s+1}} \rceil & j = J. \end{cases}$$

Then the coupling argument for perfect tries used in Section 6.1 can still be applied, now replacing $\mathbf{E}N_j$ by b_j.

In this way, a trie consisting of $Y_1, \dots, Y_n$ can be constructed, with its subtrees having fixed sizes of

$$|\{i : \ell(Y_i, y) \wedge J = j\}| = b_j. \tag{7}$$

If the original trie is good, then the number of indices i for which $X_i \neq Y_i$, called *ghosts*, is bounded by

$$\sum_{j=0}^{J-1} \alpha \times \mathbf{E}N_j + (\alpha \mathbf{E}N_J + J) = \alpha n + J.$$

A trie with these properties is called *almost perfect*.

Let T_1^* denote the number of search steps starting from node 1 via node i with $\ell(Y_i, y) < J$ in the almost perfect trie. If T_1^* and T_1' are coupled the same way as they were in Section 6.1, then $[T_1^* \neq T_1'] \subseteq B$, where B is the event that at least one node in the buckets encountered during a search is a ghost. Let A be the event that the trie is good, which has probability $o(1)$ by Lemma 2. One can check that

$$\mathbf{P}\{T_1^* \neq T_1'\} \leq \mathbf{P}\{B\} \leq \mathbf{P}\{B, A\} + \mathbf{P}\{A^c\} \leq mk(m^{-3/2} + \frac{m}{2^m}) + o(1) = o(1).$$

Again, Theorem 1 follows if

$$\frac{T_1^*}{\log_2 n} \overset{p}{\to} \frac{1}{\mu_k}, \qquad \text{as } n \to \infty.$$

7.2 Filling the Buckets with Replacement

The coupling argument used in Section 6.2 to deal the problem that buckets are filled by sampling without replacement can be adapted for an almost perfect trie. Let p_j be the probability that k items sampled without replacement from a set of size $b_{j+1} + \dots + b_J$ have conflicts. Observe that, for n large enough,

$$b_{j+1} + \dots + b_J \geq \frac{n}{2^{j+1}} - (j+1) \geq \frac{n}{2^{j+2}}.$$

Thus it follows from the union bound that

$$p_j \leq \binom{k}{2} \frac{1}{b_{j+1} + \dots + b_J} \leq \frac{k^2}{2(b_{j+1} + \dots + b_J)} \leq \frac{2^{j+1}}{n}.$$

Let the search time of sampling without replacement be T_1^{**}. Let T_1^{**} and T_1^* be coupled in the same way as they were in Section 6.2. Let D be the event that during the search an unmarked bucket is encountered. Since $[T_1^* \neq T_1^{**}] \subseteq D$, one can check that

$$\mathbf{P}\{T_1^* \neq T_1^{**}\} \leq \mathbf{P}\{D\} \leq \sum_{j=0}^{J-1} p_j < \frac{4k^2}{m^4} = o(1).$$

So once again, Theorem 1 follows if

$$\frac{T_1^{**}}{\log_2 n} \xrightarrow{p} \frac{1}{\mu_k}, \qquad \text{as } n \to \infty.$$

7.3   Analyzing T_1^{**} Using a Sum of I.I.D. Random Variables

Consider two partitions of a line segment L of length n. From left to right, cut L into $J+1$ consecutive intervals $B_0, \ldots, B_J$, with $|B_j| = b_j$, where $|a|$ denotes the length of a. Again, from left to right, cut L into infinite many consecutive intervals $B_0', B_1', \ldots$, with $|B_j'| = 1/2^{j+1}$.

Observe that for $0 \leq j < J$, B_j and B_j' do not completely match since B_j is wider than B_j'. However, since $|B_j| - |B_j'| \leq 1$, for $0 \leq j < J$, the distance between the right endpoints of B_j and B_j' is at most J. Therefore, the total length of unmatched regions, which are are called *death zones*, is $O(J^2)$.

Let $\overline{\Delta}_0, \overline{\Delta}_1, \ldots$ and $V_0, V_1, \ldots$ be the same as in Section 6.3. A coupling between them can constructed as follows: pick one point z_0 uniformly at random from the entire L. If z_0 falls in interval B_j, let $\overline{\Delta}_0 = j$. If z_0 falls in interval B_j', let $V_0 = j$. Note that $\overline{\Delta}_0 \overset{\mathcal{L}}{=} \ell(Y_1, y)$. Also note that since

$$\mathbf{P}\{V_0 = j\} = \mathbf{P}\{z_0 \in B_j'\} = \frac{|B_j'|}{n} = \frac{1}{2^{j+1}}, \qquad j = 0, 1, \ldots,$$

V_0 is geometric $(1/2)$ minus one, as desired.

Assume that $\sum_{s=0}^{t-1} \overline{\Delta}_s = j$. Pick k points from the line segment starting from B_{j+1}' to the right endpoint of L. Let $V_t = s$ such that the rightmost one of the k points falls into B_{j+s}'. Since

$$\mathbf{P}\{V_t < s\} = \mathbf{P}\{\text{all } k \text{ points are in } B_{j+1}', \ldots, B_{j+s-1}'\} = \left(1 - \frac{1}{2^{s-1}}\right)^k,$$

V_t is again the maximum of k i.i.d. geometric $(1/2)$.

If not all the k points are in the range of $B_{j+1}, \ldots, B_J$, keep picking more points until k of them are within this region. Let $\overline{\Delta}_t = s$ such that the rightmost of the these k points falls into B_{j+s}. Chosen in this way, $\overline{\Delta}_t$ has the same distribution as how much progress one makes at step t of the search. Therefore

$$T_1^{**} \overset{\mathcal{L}}{=} \inf\left\{t : \sum_{s=0}^{t} \overline{\Delta}_s \geq J\right\}.$$

It follows from Lemma 4 that if

$$T_1^{***} = \inf\left\{t : \sum_{s=0}^{t} V_s \geq J\right\},$$

then $T_1^{***}/\log_2 n \xrightarrow{p} 1/\mu_k$ as $n \to \infty$.

Let E be the event that at some step of the previous coupling, at least one of the first k chosen points falls into death zones. Note that $[T_1^{**} \neq T_1^{***}] \subseteq E$. Therefore,

$$\mathbf{P}\left\{T_1^{**} \neq T_1^{***}\right\} \leq \mathbf{P}\left\{E\right\} \leq \sum_{j=0}^{J-1} k\frac{J^2}{b_J} \leq \frac{m^3 k}{m^4 - m} = o(1).$$

So the proof of Theorem 1 when n is an arbitrary integer is complete.

8   Conclusions

In a Kademlia system, one often searches for a random ID. Although T_1 is the searching time for a fixed ID, Theorem 1 still holds if the target y is chosen uniformly at random from $\{0,1\}^d$.

If $d \sim c\log_2 n$ with $c > 2$, there is no essential difference between sampling the n IDs with or without replacement from $\{0,1\}^d$ as the probability of a collision in sampling with replacement is $o(1)$. This is the well known *birthday problem*. Since in practice, a Kademlia system hands out a new ID without checking its uniqueness, it is wise to have $c > 2$ since then a randomly generated ID clashes with any existing ID with very small probability.

Recall that $\mu_k = \sum_{j=1}^{\infty} 1 - \left(1 - 1/2^{j-1}\right)^k$. Since the terms in the sum decrease in j, μ_k can be bounded:

$$\mu_k \geq \int_0^\infty 1 - \left(1 - \frac{1}{2^x}\right)^k \mathrm{d}x = \frac{H_k}{\log 2},$$

$$\mu_k \leq \int_0^\infty 1 - \left(1 - \frac{1}{2^x}\right)^k \mathrm{d}x + 1 = \frac{H_k}{\log 2} + 1.$$

Here $\log w$ denotes the natural logarithm of w, and $H_k = \sum_{s=1}^{k} 1/s$ is the k-th harmonic number. Since $H_k \sim \log k$,

$$\lim_{k\to\infty} \frac{\mu_k}{\log_2 k} = \lim_{k\to\infty} \frac{H_k}{\log 2 \times \log_2 k} = 1.$$

Thus, $T_1/\log_k n \xrightarrow{p} \log_2 k/\mu_k = 1 + o_k(1)$. Since $T_1/(\frac{1}{2}\log_2 n) \xrightarrow{p} 1$ when $k = 1$, an increase in storage by a factor of k results in a modest decrease in searching time by a factor of $\log(k)/(2\log 2)$.

In [2], it has been proved that if $X_1 = x_1, \ldots, X_n = x_n$ for fixed $x_1, \ldots, x_n$, then

$$\sup_{x_1,\ldots,x_n} \sup_i \sup_y \mathbf{E}T_i \leq \left(\frac{\log 2}{H_k} + o(1)\right) \log_2 n.$$

Thus Theorem 1 implies that the above upper bound is not far from tight when k is large. Table 8 lists the numeric values of $1/\mu_k$ and $\log(2)/H_k$ for $k = 1, \ldots, 10$.

If $k = \Theta(\log n)$, then $T_1 \sim \log n/\log\log n$ in probability as $n \to \infty$. The proof of Theorem 1 is for fixed k only, but one can verify that only minor changes are needed to make it work for such modest increase in k as a function of n. More specifically, to make the coupling with searching in a perfect trie work, we only need to redefine $J = \min\{j : 1/2^{j+1} < m^7\}$ and $\alpha = m^{-3}$. And Lemma 4 needs to use a version of the weak law of large numbers [4, thm. 2.2.4] instead of the strong law of large numbers to deal with the fact that $\mathbf{E}V$ is not a constant anymore.

Table 1: Numeric values of $1/\mu_k$ and $\log(2)/H_k$.

k	$1/\mu_k$	$\log(2)/H_k$
1	0.5000000000	0.6931471806
2	0.3750000000	0.4620981204
3	0.3181818182	0.3780802804
4	0.2853260870	0.3327106467
5	0.2635627530	0.3035681083
6	0.2478426396	0.2829172166
7	0.2358018447	0.2673294911
8	0.2261891923	0.2550344423
9	0.2182781689	0.2450176596
10	0.2116151616	0.2366523364

If $k = n^{\Theta(1)}$, we can show that $T_1 = \Theta(1)$ in probability. Note that here only an upper bound of T_1 is needed. Assuming that the ID trie is good, it can be proved that in each search step the length of the common prefix of the current node and the target node increases by at least $c \log n$ with high probability, where c is a constant depending on k. Thus after at most $O(1)$ steps, the current node and the target node are both in a subtree of size at most k. Then the search terminates after one more step.

References

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

[2] X. Cai and L. Devroye. A probabilistic analysis of Kademlia networks. In *Algorithms and Computation*, volume 8283 of *LNCS*, pages 711–721. Springer, Berlin/Heidelberg, Germany, 2013.

[3] S. A. Crosby and D. S. Wallach. An analysis of BitTorrent's two Kademlia-based DHTs. Rice University, Houston, TX, USA. Available online, 2007.

[4] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

[5] N. Johnson, A. Kemp, and S. Kotz. *Univariate Discrete Distributions*. Wiley, Hoboken, NJ, USA, 2005.

[6] M. F. Kaashoek and D. R. Karger. Koorde: A simple degree-optimal distributed hash table. In *Peer-to-Peer Systems II*, pages 98–107. Springer, Berlin/Heidelberg, Germany, 2003.

[7] P. Maymounkov and D. Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. In *Peer-to-Peer Systems*, volume 2429 of *LNCS*, pages 53–65. Springer, Berlin/Heidelberg, Germany, 2002.

[8] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. *Theory of Computing Systems*, 32(3):241–280, 1999.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. *SIGCOMM Computer Communication Review*, 31(4):161–172, 2001.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, volume 2218 of *LNCS*, pages 329–350. Springer, Berlin/Heidelberg, Germany, 2001.

[11] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of 1st International Conference on Peer-to-Peer Computing*, pages 101–102, 2001.

[12] M. Steiner, T. En-Najjary, and E. W. Biersack. A global view of Kad. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 117–122, New York, NY, USA, 2007. ACM.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Computer Communication Review*, 31(4):149–160, August 2001. ISSN 0146-4833.

[14] W. Szpankowski. *Average Case Analysis of Algorithms on Sequences*. Wiley, Hoboken, NJ, USA, 2011.

[15] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22:41–53, 2004.