Fairness in Kademlia with Random Node Joins

Zoltán Novák, Zoltán Pap

***Abstract*—Kademlia is among the most prevalent Distributed Hash Table (DHT) protocols in practice. To understand load-balancing and fairness properties of any DHT system one of the key requirements is to study and understand the zone size distribution of the network. Already existing and well known analytical results in this field are not applicable to Kademlia directly, due to its unique addressing mechanism. We show that a direct connection exists between the size of the zones of a given Kademlia network and the shape parameters of the data structure called PATRICIA trie filled with the overlay addresses of the same network. Then analytical description of the asymptotic properties of the Kademlia zone size distribution is provided based on the existing literature on random binary tries. We compare Kademlia to the Chord DHT, and show that Kademlia provides a fairer zone size distribution. These results can be used to achieve better load balancing in DHT systems.**

***Index Terms*—Consistent hashing, load balancing, asymptotic bounds, peer-to-peer networks.**

I. INTRODUCTION

We examine load distribution in the Kademlia [1] distributed hash table (DHT) system which is one of the most widely used peer to peer (P2P) overlay in practice in these days[1].

Distributed hash tables [1]–[3] - as their name suggest - are for storing and retrieving arbitrary data in P2P networks using hash keys. Data is distributed among all participant peers in the network. Each node is responsible for a given part of the hash space called zone. A node stores a given value if the hash key of the value falls into its zone. The zone of the node is usually determined by a predefined relation between the overlay address of the node and the addresses of the other online nodes. The goal is to minimize the number of zones that have to be modified (increased or decreased) when additional nodes join or leave the system. A somewhat contradicting goal is to keep the zone sizes balanced. These goals are usually achieved by a technique called consistent hashing [4].

The relative size of the zone of a given node determines the expected relative number of data items it has to store. It also influences the number of routes directed through the node. Thus having a larger zone size means a larger expected average load for the node in the system.

Using the assumption that both the node addresses and the hash keys are uniformly distributed[2], the zone size distribution of a DHT protocol can be calculated by using probabilistic models.

Manuscript received April 9, 2015, revised June 12, 2015

Zoltán Novák and Zoltán Pap are with Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 1117, Budapest Magyar tudósok krt 2, Hungary. E-mail: {novak,pap@tmit.bme.hu}

[1]See http://en.wikipedia.org/wiki/Kademlia#Implementations,

[2]The first is true by definition, because in DHTs each joining node choose a random address. The second part can be considered true due to the properties of the commonly used hash functions.

The exact distribution can be used to describe, compare or evaluate the performance characteristics of different DHT systems, or to devise efficient uniform random node selection algorithms [5], [6]. These algorithms could be used for statistical estimations in large networks or as an algorithmic building block in randomized network algorithms. Uniform random node selection can also be used directly for load balancing purposes. A specific example of application was presented by Scott Lewis et al. [7]. Their scalable Byzantine agreement algorithm is based on the availability of uniform random node selection in a network.

Load balancing is also one of the areas that could benefit much from the exactly known apriori distribution of the load [8].

II. RELATED WORK

A. Consistent hashing

Karger et al. [4] have been introduced consistent hashing to minimize the number of values that have to be moved upon a hash table resize. They have provided the following algorithm:

The storage nodes (buckets) are randomly placed (hashed) onto a unit circle, and each bucket stores the data with hash keys between its hash and the hash of the previous bucket. The handling of the hash space in the Chord [2] DHT system is based upon the same concept.

B. DHT zone distribution

The probabilistic properties of the zone sizes have been investigated in the original article that has introduced cosnsitent hashing [4]. The load distribution of Chord DHT have been first investigated by using simulation in [2]. The asymptotic distribution of the minimal zone size in Chord have been described in [5]. Cuevas et al. [9] have examined routing fairness in the Chord system, based on the Chord zone size distribution (as nodes with larger zone tend to appear in other nodes routing table more frequently).

Finally, Wang et al. [10] have provided several limits – that are true with high probability – for Chord's zone size distribution, such as the distribution of minimal, maximal zones and have also examined different joining strategies like half splitting[3], and multipoint sampling[4]. It may be interesting to mention, that for the half-splitting case, the authors of [10] have relied on results coming from the research of regular tries, but have not recognised the connection between the Kademlia address space and PATRICIA tries. This article also contains a good collection of the prior results of the field.

[3]When a joining node always choose an address that splits the original zone into two equal parts.

[4]When a joining node choose its address by splitting the largest zone it has found after sampling some random locations.

C. Kademlia

For Kademlia the analytical literature is sparse. Recently Cai and Devroye [11] have provided analytical results about the search times in Kademlia. Their method is based on regular tries: they've started with the initial assumption that the address trie is balanced (in this case regular and PATRICIA tries are similar), and then have refined the result by relaxing on this balancedness assumption. They did not recognise the direct connection between these exact (unbalanced) description of the address space and PATRICIA tries.

D. Occupancy problem, poissanization

As we will show in section IV, the zone size of a Kademlia node depends on whether particular address sets – these sets are depend on the address of the given node – contain at least one online node. The general version of this problem is called occupancy problem:

Given n balls and c cells we assign the balls into the cells randomly. What will be the probability that exactly k cells remain empty?

In the most general case, cells can have different selection probabilities, or the number of cells can be infinite – the only restriction then is that the sum of the selection probabilities have to be 1. This field of probability theory [12]–[14] gives a general theoretical framework for our problem.

It is also worth to mention a general technique to solve similar problems: analytic depoissonization [15]. In many cases these kind of balls-in-urns problems can be modelled easier with poissonization. Poissonization means that the exact Bernoulli models are replaced with approximative Poisson models (e.g. imagine balls arriving into the urns according to Poisson processes). With depoissonization it is possible to translate back the results of the Poisson model to the original Bernoulli model.

Reference [13] has also presented results about the equivalence of the moments of the original and the poissonizated occupancy distributions.

E. PATRICIA tries

PATRICIA trie [16] is the compact version of the regular trie (also called prefix tree). These structures are commonly used to efficiently store strings together with their prefixes. By providing efficient prefix search, they are particularly suitable for storing dictionaries or routing tables.

In the generic trie each node of the tree represents a character of the stored string, and a path to an internal node in the tree represents a prefix string. Below that node one can find all the strings sharing that same prefix. A path form the root to a leaf node gives a stored string, where in each step we get the next character of the string.

PATRICIA trie (also called radix tree) is a space optimized version of the regular trie, where each node with only one child is merged with its parent. In this case a node can contain larger fragments of the prefix not just one character. (Figure 1 shows a PATRICIA trie storing five binary strings.)

As we show later, the shape parameters of the binary PATRICIA trie are directly related to the zone size distribution of the Kademlia DHT system.

Unfortunately it has been proved to be notoriously hard to describe the exact shape parameters of random PATRICIA tries, and despite it has been introduced for more than forty years ago, the properties of the PATRICIA trie are still actively researched. Luckily there exists many asymptotic results in this field, that can be applied directly to our problems.

References [17]–[19] provide asymptotic and limiting distributions of various shape parameters of random PATRICIA tries. A recent result about the expected value of the number of tree nodes at a certain level of the trie have been presented in [20]. An interesting result is that the variance of the insertion cost of random strings into PATRICIA tries – which is related to the path length distribution – is constant: $1 + \mathrm{O}(1)$ $(= 1.000000000001237\ldots)$ [21]. Finally there are also results about the asymmetrical case where the input alphabet is not uniformly distributed [22], [23].

III. KADEMLIA

This section is a short introduction to the Kademlia [1] DHT.

Kademlia is based on a 160 bit address space, to which both nodes and keys are mapped. Each key-value pair is stored on the node having the closest overlay address in the system to the given key. Distance is calculated using the result of bitwise binary XOR operator ($\oplus$) interpreted as a natural number:

$$\mathrm{d}(nodeaddress, key) = nodeaddress \oplus key$$

Each node maintains 160 tables to store routing information. In Kademlia terminology these tables are called k-buckets. The i-th k-bucket contains at most K nodes whose distance from the current node is between 2^{160-i} and $2^{160-i+1}$, where K is a pre-chosen system parameter.

K-buckets are ordered lists of nodes, with the most recently seen node at the beginning of the list. If a node A receives a message from another node B, than A tries to insert B into the appropriate k-bucket, if there's still room. If the given k-bucket is full, A sends PING to the node from the end of the list; if it replies, A moves it to the head of the list; if it does not, A deletes it from the list, and replaces it with B. With adequate network traffic, k-buckets remain consistent thanks to the procedures above.

A. Searching

The Kademlia protocol defines four remote procedure calls (RPC). Each participating node have to implement these:

- PING, check if a node is still connected;
- STORE, stores a key and corresponding data;
- FIND_NODE with an address as its parameter, returns the K closest values to the given address from the node's routing tables;
- FIND_VALUE with a key as its parameter, if a node stores data corresponding to the key, it returns the result data; otherwise it behaves identically to FIND_NODE.

Using these RPCs a node can find the closest peer to a given address. For example lets assume that a node (X) wants to

look up the closest node to a key (y). The search goes through the following steps:

1) Node X creates a list L containing the K closest addresses to y. Initially it fills this list from its own k-buckets.
2) X selects α unmarked nodes from the list, and runs the FIND_NODE RPC on them (α is a predefined system-wide parameter).
3) X updates the list L by merging the return values of the FIND_NODE RPCs. Then it keeps only the K closest addresses to y. It also marks every node in the list on which the FIND_NODE RPC has been already run.
4) If the list still contains unmarked nodes, return to step 2.
5) The result node is the one with the closest address to y in L.

When a node leaves the network, it simply copies its data to the nearest node, and disconnects.

B. Defining zone distribution

Definition 1: Let $A = \{0, 1, \ldots, 2^{160} - 1\}$ be the set of all addresses in the system. Let N be the set of occupied addresses (the set of online nodes). We denote the number of online nodes $|N|$ with n.

Joining nodes choose a uniformly distributed random address independently from each other. We assume that $n > 0$, i.e., every system has at least one node online.

Definition 2: Let $X \in N$ be the address of a node in the system, then define $\mathrm{Close}(X) \subseteq A$ as the set of addresses where:

$$\mathrm{Close}(X) = \{Z \mid \forall Y \in N, Y \neq X, X \oplus Z < Y \oplus Z\}$$

where $\oplus$ is the bitwise XOR operation, and its result is interpreted as a natural number.

$\mathrm{Close}(X)$ can be imagined as a kind of Voronoi cell of X: the set of all addresses that are closer to X – according to the XOR distance – than to any other online node.

Definition 3:

Let the zone size $\mathrm{T}(X)$ of a node $X \in N$ be:

$$\mathrm{T}(X) = \frac{|\mathrm{Close}(X)|}{|A|},\ (0 < \mathrm{T}(X) \leq 1).$$

The zone size of X represents the portion of addresses (A) that are closer to X than to any other online node. For example if $\mathrm{T}(X) = 0.5$, then if a uniformly random address R from the address set A is chosen – this is the case in practice when a hash key is calculated to store a value – the closest online node to R will be X with a probability of 0.5.

This way the distribution of the zone sizes in the system corresponds to the load distribution – assuming that the hash keys are uniformly distributed.

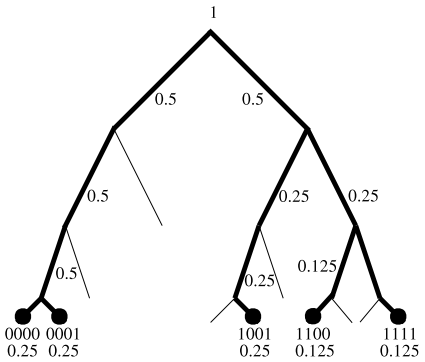
IV. KADEMLIA AND PATRICIA TRIES

In this section we present a connection between the Kademlia zone sizes and the shape of the PATRICIA trie of the Kademlia addresses. Then the cumulative distribution function of Kademlia zone sizes is provided. Utilizing existing results about random binary PATRICIA tries we describe the first two moments of this zone size distribution, and an estimation of the minimal zone size in the system is also provided. Finally we provide an asymptotic estimation of the Jain's fairness index of the zone distribution, as a measure of load fairness.

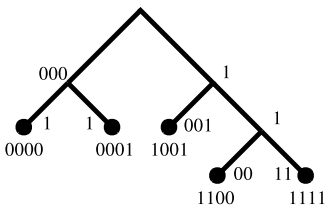
A. Visualizing Kademlia zone sizes

First let us try to visualise the zone size distribution in XOR distance as defined in section III-B.

1) The sum of zone sizes for all online nodes in the system is 1.
2) The sum of zone sizes for online nodes whose addresses start with 0 or 1 are 0.5 and 0.5 respectively, if there is at least one online node in both the 0xxx... and the 1xxx... address space.
3) Groups of nodes with prefix: 00, 01, 10, 11 share 0.25, 0.25, 0.25, 0.25 parts of the whole territory if all address prefixes contain at least one online node.
4) and so on...



(a) Division of zone sizes in XOR distance



(b) Kademlia addresses in PATRICIA trie

Figure 1: Visualizing Kademlia zone sizes

What happens if there isn't any node with prefix 10? Then nodes with prefix 11 will share on a 0.5 territory, as they are the only nodes with address prefix 1, and nodes with prefix 1 share 0.5 territory according to bullet 2 above. In Figure 1a the division of the zones is depicted for addresses: 0000, 0001, 1001, 1100, 1111 in the four bit address space.

B. Kademlia zone sizes and the PATRICIA trie of addresses

We have seen that according to the XOR distance division of the zone happens only if there is a branching in the regular

trie of the addresses. Because in PATRICIA trie the internal nodes with only one child (non-branching nodes) are merged with their parents, the path length of the trie to a given address leaf equals the number of branching on the path in the original trie.

Using this insight the division of the zones can be described with a corresponding PATRICIA prefix trie. The leaves of the trie are the occupied addresses. At the root, we begin with a zone size of 1, and at every lower level the territory is divided by two.

Figure 1 shows a division of zones between nodes: 0000, 0001, 1001, 1100, 1111 in the four bit address space, and the corresponding PATRICIA trie (in Figure 1b) storing the same addresses.

The size of the zone of a node is 2^{-l} where l is the path length of the address in the binary PATRICIA trie.

The sum of the zone sizes is always one – in accordance with the well known Kraft's inequality, that states that for every binary tree:

$$\sum_{\ell \in \text{leaves}} 2^{-\text{depth}(\ell)} \leq 1$$

where equality holds if every internal node has two children, which is true by definition in PATRICIA tries.

C. Distribution of zone sizes in Kademlia

We have two conflicting assumptions:

1) Joining nodes choose their addresses from a finite address space independently and randomly with uniform distribution, meaning that address collision is possible
2) Every node has a unique address

If there would be any measurable chance of address collision, it could be handled by increasing the address space. Simply we assume that this method would be used instead of other methods such as reconnection with a different address. It is worth considering that although the analytical description would be somewhat different for the aforementioned two cases, the practical numerical values would only be different by the order of magnitude of the hash collision probability - which is – by design – negligible in practice. Therefore in the rest of the paper we simply consider the size of the address space (the height of the PATRICIA trie of addresses) unbounded. Similar simplifications (for example modelling the Chord ring with a continuous unit circle) are prevalent in the literature.

Definition 4: Let $\mathrm{P}_n(\mathrm{T} \leq x) = \mathrm{F}_n^T(x)$ be the cumulative distribution function (CDF) of T (the zone sizes) in a system with n independently and randomly chosen node addresses. Then, in a system with one node ($n = 1$):

$$\mathrm{F}_1^T(x) = \begin{cases} 0 & \text{if } x \leq 1, \\ 1 & \text{if } x > 1. \end{cases}$$

We can write a recursive definition of $\mathrm{F}_n^T(x)$ using the law of total probability:

1) Let us visualise the n node addresses at the root of the corresponding regular trie (Fig. 1a). Every address begins with 0 or 1 with a probability of 0.5 respectively. The root node divides the n nodes into two sets.
2) The cardinality of these two sets has a binomial distribution, and they sum up to n:

$$P(\text{no address begins with } 0) = \binom{n}{0} 2^{-n},$$
$$P(1 \text{ address begins with } 0) = \binom{n}{1} 2^{-n},$$
$$\vdots$$
$$P(n \text{ addresses begin with } 0) = \binom{n}{n} 2^{-n}.$$

3) Let us assume that 5 addresses begin with 0 and $n-5$ with 1. If $\mathrm{F}_5^T(x)$ and $\mathrm{F}_{n-5}^T(x)$ is known, the CDF of their combination can be written. As each of the two branches shares upon only 0.5 zone, we have to use $\mathrm{F}_5^T(2x)$ and $\mathrm{F}_{n-5}^T(2x)$. Note that this would not be the case if the division was $(n;0)$ or $(0;n)$, because than the zone is not halved (no new level added in the corresponding PATRICIA trie)!
4) Using the law of total probability we can write the following recursive definition:

$$\mathrm{F}_n^T(x) = \frac{1}{2^n} \left(\binom{n}{0} F_n^T(x) + \binom{n}{n} F_n^T(x) + \right.$$
$$\left. + \sum_{k=1}^{n-1} \binom{n}{k} \left(\frac{k}{n} \mathrm{F}_k^T(2x) + \frac{n-k}{n} \mathrm{F}_{n-k}^T(2x) \right) \right)$$

5) Finally – after rearranging occurrences of $\mathrm{F}_n^T(x)$ to the left – we reach the following recursive formula as the CDF of the exact zone distribution of the Kademlia DHTs:

$$\mathrm{F}_1^T(x) = \begin{cases} 0 & \text{if } x \leq 1, \\ 1 & \text{if } x > 1. \end{cases}$$

$$\mathrm{F}_n^T(x) = \frac{1}{2^n - 2} \sum_{k=1}^{n-1} \left(\binom{n-1}{k-1} \mathrm{F}_k^T(2x) + \right.$$
$$\left. + \binom{n-1}{k} \mathrm{F}_{n-k}^T(2x) \right)$$

D. Moments of the zone sizes in Kademlia

Starting from the known recursive generating function of the PATRICIA trie's path lengths:

$$g_x(1) = 1$$

$$g_x(n) = \frac{2x}{2^n - 2} \sum_{i=1}^{n-1} \binom{n-1}{i-1} g_x(i)$$

We can recognise that by substituting $x = \frac{1}{2}$ to this generating function of the trie path length distribution, we get the formula for the expected value of the Kademlia zone sizes.

Here the result can be inferred much easier considering the fact that nodes share the entire hash space, so on average they get nth part of it:

$$E(T) = \frac{1}{n} \ (= g_{\frac{1}{2}}(n) = \frac{1}{2^n - 2} \sum_{i=1}^{n-1} \binom{n-1}{i-1} g_{\frac{1}{2}}(i))$$

As a by-product this short-cut provides an interesting identity for the recursive formula of the generating function.

Using the similar insight about the generating function, the variance of the zones can be defined as:

$$V_n(T) = E_n(T^2) - E_n^2(T) = g_{\frac{1}{4}}(n) - \frac{1}{n^2}$$

We can analyse the asymptotic behaviour of $g_{\frac{1}{4}}(n)$ by using the generating function of the poissonizated limiting distribution:

$$\mathrm{G}_x(n) = \prod_{i=1}^{\infty} \left[e^{\frac{-n}{2^i}} + (1 - e^{\frac{-n}{2^i}})x \right]$$

This distribution is derived according to section II-D. We simply assume that branching happens at a given node in PATRICIA trie according to Poisson distribution instead of using the proper Binomial distribution.

According to [14] the variance of the original and the poissonizated distribution is asymptotically close. For certain cases the difference is $\mathrm{o}(1)$ (small ordo), meaning that they are asymptotically equivalent[5]. Therefore we calculate $G_{\frac{1}{4}}(n)$ first.

From the generating function we have:

$$G_{\frac{1}{4}}(n) = \prod_{i=1}^{\infty} \left[e^{\frac{-n}{2^i}} + (1 - e^{\frac{-n}{2^i}}) \frac{1}{4} \right] = \prod_{i=1}^{\infty} \left[\frac{1}{4} + \frac{3}{4} e^{\frac{-n}{2^i}} \right]$$

As a simplification let's assume that n is a power of two, then the limit of $G_{\frac{1}{4}}(n)$ as n goes to 2^∞ is:

$$\lim_{n \to 2^\infty} G_{\frac{1}{4}}(n) = \lim_{n \to 2^\infty} \prod_{i=1}^{\infty} \left[\tfrac{1}{4} + \tfrac{3}{4} e^{\frac{-n}{2^i}} \right] =$$

$$= \lim_{n \to 2^\infty} \frac{1}{4^{\log_2 n}} \prod_{i=1}^{\log_2 n} \left[1 + 3e^{\frac{-n}{2^i}} \right] \prod_{i=\log_2 n+1}^{\infty} \left[\frac{1}{4} + \frac{3}{4} e^{\frac{-n}{2^i}} \right] =$$

$$= \lim_{n \to 2^j, j \to \infty} \frac{1}{n^2} \prod_{i=1}^{j} \left[1 + 3e^{-2^{j-i}} \right] \prod_{i=j+1}^{\infty} \left[\frac{1}{4} + \frac{3}{4} e^{-2^{j-i}} \right] =$$

$$= \lim_{n \to 2^j, j \to \infty} \frac{1}{n^2} \prod_{i=0}^{j-1} \left[1 + 3e^{-2^i} \right] \prod_{i=1}^{\infty} \left[\frac{1}{4} + \frac{3}{4} e^{-2^{-i}} \right] =$$

$$= \lim_{n \to 2^\infty} \frac{1.5254695585786\ldots}{n^2}$$

The constant of the last line is the result of calculating the (existing) limits of the products numerically. By relaxing the assumption that n is a power of two, the results may be different due to the nonzero fractional part of $\log_2 n$.

Instead of deriving this more generic solution, we have simply used this specific asymptotic behaviour of the limiting distribution at large powers of 2 as a clue to search for $g_{\frac{1}{4}}(n)$ in the form of: $g_{\frac{1}{4}}(n) \simeq c/n^2$.

By numerical calculations we have found that c is oscillating around 1.525 with a decreasing amplitude as n increases[6]:

$$g_{\frac{1}{4}}(n) \simeq \frac{1.525}{n^2}$$

This gives:

$$V_n(T) = g_{\frac{1}{4}}(n) - \frac{1}{n^2} \simeq \frac{0.525}{n^2}$$

An alternative characterization of the zone size distribution can be given by calculating Jain's fairness index. This index can be used to describe fairness with a constant value, when the participants share on some finite resource (such as the hash space in our case). It has been defined as:

$$\mathcal{J}(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i \right)^2}{n \sum_{i=1}^{n} x_i^2}$$

The result ranges from $1/n$ when one node gets all the resources (worst case) to 1 when nodes have equal shares of the resources (best case). The index is k/n if k nodes equally share the resource, while the other $n - k$ node receive zero amount.

From the variance calculation it follows that in the case of Kademlia:

$$\mathcal{J}_T \approx 1/1.525 \approx 0.655$$

E. Distribution of the size of the minimal zone in Kademlia

The distribution of the minimal zone can be used to achieve efficient uniform random node selection in a Kademlia system. The distribution of the minimal zones can be derived with the method presented in section IV-C.

For the details of the derivation of the exact cumulative density function (CDF) of the minimal sized zone in Kademlia, please refer to the previous work of the authors [6]. Here we present the result without further explanation:

$$\mathrm{F}_1^{T_{min}}(x) = \begin{cases} 0 & \text{if } x \leq 1, \\ 1 & \text{if } x > 1. \end{cases}$$

$$\mathrm{F}_n^{T_{min}}(x) = \frac{1}{2^n - 2} \sum_{k=1}^{n-1} \binom{n}{k} \Big(\mathrm{F}_k^{T_{min}}(2x) + \mathrm{F}_{n-k}^{T_{min}}(2x) - \\ - \mathrm{F}_k^{T_{min}}(2x) \mathrm{F}_{n-k}^{T_{min}}(2x) \Big)$$

The minimal zone size in Kademlia is in direct relationship with the height of the PARTICIA trie of addresses. Having a trie with height h, the corresponding Kademlia network will have a minimal zone size of: 2^{-h}.

Key results about the heights of PATRICIA tries have been presented in [18]. For the random binary case the height of the

[5]It may be interesting to mention here that $G_{1/2}(n)$ – the expected value of the poissonizated distribution – can be given exactly in closed form, it equals: $\frac{1-e^{-n}}{n}$

[6]After $n > 1000$ these four decimal digits of 1.525 gets stabilized.

trie is – depending on n – oscillating around the most probable value of:

$$h_1 = \lfloor \log_2 n + \sqrt{2 \log_2 n} - \frac{3}{2} \rfloor + 1$$

The height of the trie is concentrated on h_1 for most n, and for some n it is either concentrated on h_1 and $h_1 + 1$ or on h_1 and $h_1 - 1$.

From this the most probable value of the minimal zone is simply:

$$T_{min} = 2^{-h_1}$$

The random node selection algorithm of the authors [6] relies on the estimation of minimal zone size in the system. In that particular case using an estimate that is lower than the actual minimal zone size results in a perfectly uniform random node selection. Contrarily underestimating the minimal zone size by a large margin results in a large increase in the run time of the random node selection algorithm. In this special case the probable underestimation of the actual minimal zone size:

$$2^{-h_1 - 1} = 2^{-\lfloor \log_2 n + \sqrt{2 \log_2 n} - \frac{3}{2} \rfloor} \gtrapprox T_{min}$$

could provide a viable trade-off.

Alternatively by using the asymptotics of [20][7], any zone size can be characterized with the expected number of nodes having smaller zone – this can be useful to estimate a (minimal) zone together with the known expected number of outliers.

V. COMPARISON TO CHORD

Some of the basic properties of Chord zone distribution – derived from the model of random points on unit circle – are summarized in Table I for comparison purposes.

Zone size	Chord	Kademlia
Average	$1/n$	$1/n$
Variance	$(n-1)/(n^2+n^3)$	$\approx 0.525/n^2$
Minimal	$1/n^2 (= 2^{-2 \log_2 n})$	$\geq 2^{-\lfloor \log_2 n + \sqrt{2 \log_2 n} - \frac{3}{2} \rfloor}$
Jain's fairness	$0.5 + 1/n$	0.655

Table I: Main parameters of the zone size distributons

The full derivation of these results are available in the literature (section II-B). Only a small summary is presented here for comparison purposes.

Cumulative Density Function (CDF) of zone sizes in Chord:

$$\mathrm{P_n}(T \leq x) = \mathrm{F_n}(x) = 1 - (1-x)^{n-1} \ (0 \leq x \leq 1)$$

Probability density function (PDF) of zone sizes in Chord:

$$\mathrm{f_n}(x) = \mathrm{F_n}'(x) = (n-1)(1-x)^{n-2} \ (0 \leq x \leq 1)$$

Expected value (average zone size), as nodes share the whole hash space this result is the same as for Kademlia:

$$\mathrm{E}_n(x) = 1/n \ (= \int_0^1 x(n-1)(1-x)^{n-2} \, dx)$$

[7]The expected number of nodes at a given level of the PATRICIA trie

Variance of zone sizes:

$$\mathrm{V}_n(x) = \int_0^1 \left(x - \frac{1}{n}\right)^2 (n-1)(1-x)^{n-2} \, dx =$$

$$= \left[\frac{(n-1)(1-x)^n \left(1 - 2x + x^2 n^2\right)}{n^2(n+1)(x-1)} \right]_0^1 = \frac{n-1}{n^2+n^3}$$

This is approaching $1/n^2$ for large n, so Kademlia have a constant factor advantage here.

It follows that the Jain's fairness index of the zone sizes in Chord is:

$$\mathcal{J}_T = \frac{1}{n^2 \left(\frac{n-1}{n^2+n^3} + \frac{1}{n^2} \right)} = \frac{1}{2} + \frac{1}{2n}$$

This is 0.5 asymptotically, which is less (worse) than the result of Kademlia (0.655).

Cumulative Density Function (CDF) for the minimal zone in Chord:

$$\mathrm{P}_n^{\min}(T_{\min} \leq x) = \mathrm{F}_n^{\min}(x) = 1 - (1-nx)^{n-1} \ (0 \leq x \leq \frac{1}{n})$$

Probability density function (PDF) for the minimal zone size in Chord:

$$\mathrm{f}_\mathrm{n}^{\min}(x) = \mathrm{F}_\mathrm{n}^{\min'}(x) = (n^2-n)(1-nx)^{n-2} \ (0 \leq x \leq \frac{1}{n})$$

Expected value of the minimal zone size in Chord:

$$\mathrm{E}_n^{\min}(x) = 1/n^2 \, (= \int_0^{\frac{1}{n}} x(n^2-n)(1-nx)^{n-2} \, dx)$$

Variance of the minimal zone size in Chord:

$$\mathrm{V}_n^{\min}(x) = \int_0^{\frac{1}{n}} \left(x - \frac{1}{n^2}\right)^2 (n^2-n)(1-nx)^{n-2} \, dx =$$

$$= \frac{n-1}{n^4+n^5}$$

For the maximal zone size distribution refer to the asymptotic results of Darling [24]. We have not provided results for the maximal zone size distribution of Kademlia.

VI. CONCLUSIONS

A direct connection between Kademlia zone size distribution and the shape parameters of a PATRICIA trie built from the overlay addresses of the DHT have been presented. By relying on existing literature on random symmetric binary PATRICIA tries, we have derived some of the key parameters of the zone size distribution of Kademlia DHT.

The exact distribution of zone sizes and minimal zone sizes of Kademlia have been provided. We have also provided approximative characterisation of the moments of the zone size distribution based on the literature on PATRICIA tries.

By comparing the zone size distribution of Kademlia and Chord, we have concluded that Kademlia zones are distributed more uniformly. Zone sizes have a smaller deviation by a constant factor, and the minimal zone size is larger compared to

Chord. Kademlia zone size distribution is also fairer compared to Chord's by the measure of Jain's fairness index.

The consequence is that in general cases Kademlia achieves a more fair distribution of data than Chord, and this suggests that it may show a more uniform distribution of routing load too. Finally the results about the size of the minimal zone in Kademlia open the possibility to improve upon the random node sampling algorithm of the authors [6].

Zoltán Novák received an M.Sc. degree in software engineering at Budapest University of Technology and Economics (BME) in 2006. Between 2006 and 2010 he was a Ph.D. student at BME department of Telecommunications and Media Informatics. His main research topics are peer to peer systems and overlay networking. Since 2011 he has been working at Ericsson Hungary. Currently he is pursuing Ph.D. degree under the supervision of Zoltán Pap.

Zoltán Pap received an M.Sc. degree in Electrical Engineering at Budapest University of Technology and Economics (BME) in 2000, and an M.Sc. degree in Business Administration at Corvinus University of Budapest in 2002. From 2000 to 2006 he worked on various research fields including telecommunication networks and protocols, grid computing, peer to peer networks, model-based software development and testing at BME department of Telecommunications and Media Informatics. He received a Ph.D degree at BME in 2006. Since 2007 he has been working at Ericsson initially as systems engineer later as product manager and functional manager on several products such as the Telecom Server Platform (TSP), Ericsson's Operations Support System Radio and Core (OSS-RC) and the Smart Services Router (SSR).

REFERENCES

[1] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, (London, UK), pp. 53–65, Springer-Verlag, 2002.

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

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*, vol. 31, pp. 161–172, ACM Press, October 2001.

[4] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web," in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, (New York, NY, USA), pp. 654–663, ACM, 1997.

[5] V. King, S. Lewis, J. Saia, and M. Young, "Choosing a random peer in chord," *Algorithmica*, vol. 49, no. 2, pp. 147–169, 2007.

[6] Z. Novák and Z. Pap, "Random node sampling in kademlia," in *6th International ICST Conference on Broadband Communications, Networks, and Systems*, IEEE, 11 2009.

[7] C. Scott and L. J. Saia, "Scalable byzantine agreement," tech. rep., 2004.

[8] M. D. Mitzenmacher, *The power of two choices in randomized load balancing*. PhD thesis, 1996. Chair-Alistair Sinclair.

[9] R. C. Rumín, M. Urueña, and A. Banchs, "Routing fairness in chord: Analysis and enhancement.," in *INFOCOM*, pp. 1449–1457, IEEE, 2009.

[10] X. Wang and D. Loguinov, "Load-balancing performance of consistent hashing: asymptotic analysis of random node join," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 4, pp. 892–905, 2007.

[11] X. S. Cai and L. Devroye, "A probabilistic analysis of kademlia networks," in *Algorithms and Computation*, pp. 711–721, Springer, 2013.

[12] A. Gnedin, B. Hansen, J. Pitman, *et al.*, "Notes on the occupancy problem with infinitely many boxes: general asymptotics and power laws," *Probability surveys*, vol. 4, pp. 146–171, 2007.

[13] T. Emigh, "On the number of observed classes from a multinomial distribution," *Biometrics*, pp. 485–491, 1983.

[14] H.-K. Hwang and S. Janson, "Local limit theorems for finite and infinite urn models," *The Annals of Probability*, pp. 992–1022, 2008.

[15] P. Jacquet and W. Szpankowski, "Analytical depoissonization and its applications," *Theoretical Computer Science*, vol. 201, no. 1-2, pp. 1–62, 1998.

[16] D. R. Morrison, "Patricia - practical algorithm to retrieve information coded in alphanumeric.," *J. ACM*, vol. 15, no. 4, pp. 514–534, 1968.

[17] B. Rais, P. Jacquet, and W. Szpankowski, "A limiting distribution for the depth in patricia tries," 1990.

[18] C. Knessl and W. Szpankowski, "Limit laws for the height in patricia tries," *Journal of Algorithms*, vol. 44, no. 1, pp. 63–97, 2002.

[19] L. Devroye, "Laws of large numbers and tail inequalities for random tries and patricia trees," *Journal of Computational and Applied Mathematics*, vol. 142, no. 1, pp. 27–37, 2002.

[20] A. Magner, C. Knessl, and W. Szpankowski, "Expected external profile of PATRICIA tries," in *2014 Proceedings of the Eleventh Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2014, Portland, Oregon, USA, January 6, 2014*, pp. 16–24, 2014.

[21] H. Prodinger, "Compositions and patricia tries: no fluctuations in the variance," SODA, 2004.

[22] J. Bourdon, "Size and path length of patricia tries: dynamical sources context," *Random Structures & Algorithms*, vol. 19, no. 3-4, pp. 289–315, 2001.

[23] L. Devroye, "A study of trie-like structures under the density model," *The Annals of Applied Probability*, pp. 402–434, 1992.

[24] D. A. Darling, "On a class of problems related to the random division of an interval," *Ann. Math. Statist.*, vol. 24, pp. 239–253, 06 1953.