Ring-like DHTs and the Postage Stamp Problem

Mahadev Konar and Alexander E. Mohr
Department of Computer Science
Stony Brook University
{dev,amohr}@cs.stonybrook.edu

Abstract

We explore Chord-like ring structures for Distributed Hash Tables (DHTs) and show that the Postage Stamp Problem (PSP) is equivalent to finding optimal structures for such ring topologies. We first leverage known results for that problem to design structures for ring-like DHTs that use shortest-path routing. We then describe a variant of the PSP that corresponds to ring-like DHTs that use greedy routing and develop an algorithm that smoothly trades off between the number of finger pointers and network diameter. We include a matrix of the number of nodes that can be supported as a function of the number of finger pointers and the network diameter and also note an interesting link to the Fibonacci sequence. Finally, we see that our ring constructions can support networks of 165 million nodes compared to only 1 million nodes for Chord-like rings when both approaches use 20 finger pointers and a network diameter of 20.

Keywords: Distributed Hash Table, Postage Stamp Problem, network diameter, finger pointers.

Contact:

Name:	Mahadev Konar
Phone:	631 632 1857
email:	dev@cs.sunysb.edu
Address:	700 Health Science Drive H1121 Ay Chapin Apartments Stony Brook NY-11790

Pages: 7

Consideration for Brief Announcement: If not accepted please consider this regular presentation for brief announcement.

Consideration for Best Student Paper Award: The primary author of this paper is a full-time student whose contribution to this paper was significant.

1 Introduction

To overcome scalability constraints, various Distributed Hash Table (DHT) schemes (Chord/CAN/Pastry) [11, 7, 9] have been proposed in different p2p systems. In DHT schemes each node stores objects that correspond to a certain portion of the key space and uses a routing table to forward the request for an object not belonging to its key space to the appropriate next hop node. The goal is to locate the object in fewest number of hops while minimizing the size of the routing table.

DHTs have two primary characteristics of interest:

- *degree* or *routing table size*, the number of neighbors that a node has, and
- *hop count* or *network diameter*, the number of hops needed to reach any node from another.

Some DHT schemes are based on rings, such as Chord, and have a routing table size of $O(log_2 n)$ and a network diameter of $O(log_2 n)$. Others are based on hypercubes, such as CAN, and have a routing table of size d and a diameter $O(n^{1/d})$. Still others are based on De Bruijn graphs and achieve a routing table of size $O(1)$ with diameter $O(log_d\ n)$.

In this paper, we focus exclusively on circulant graph structures that correspond to rings such as Chord, in which all of the nodes have finger pointers with identical offsets. We explore these structures because they are easy to realize in practice: a node's finger offsets are independent of its location in identifier space, so it need not discover that information before finding its neighbors. Also, as was shown by Gummadi et al. [2], ring-like structures have very good resilience properties even when large fractions of the nodes fail.

We explore how construct good ring-like network structures given number of finger pointers and network diameter. We first formulate the problem of constructing ring-like structures when shortest-path routing is used as an optimization problem that corresponds to the Postage Stamp Problem (PSP), which allows us to apply the extensive body of work on PSP to our problem. Since there is no known polynomial time solution for PSP, finding optimal structures for arbitrary routing table sizes and network diameters is intractable for such circulant graphs.

We then consider a variant of the PSP that applies to circulant graphs when greedy routing, rather than shortest-path routing is used when choosing the next finger pointer to take. For this variant of the PSP, we develop an algorithm that calculates the maximum network size possible for a given network diameter and routing table size. We show that using a routing table of size 20 with a network diameter of 20, we can construct a network of 165 million nodes in comparison to a network of 1 million nodes using Chord.

In Section 2, we discuss background and present the relation between the Postage Stamp Problem and ring-like structures. In the background section we also present some known results for the Postage Stamp Problem and their implications to optimal ring-like DHT constructions. In Section 3, we present our algorithm. In Section 4, we show our results and compare to Chord. Section 5 concludes the paper.

2 Background

In DHT schemes, a routing algorithm is characterized by the routing table at each node. We provide the routing table formulation in terms of graphs for a better understanding of our formulation in terms of the Postage Stamp Problem.

Let G(n; $s_1, s_2, ..., s_k$) be the network with n nodes, labeled with integers modulo n and k links per vertex such that each node i is adjacent to k other nodes $i + s_1$, $i + s_2$, $i + s_3$, ..., $i + s_k$ (mod n). This kind of network is called a **Circulant Graph** [1]. Chord has s_i as

$n/2^i$ where $1 \leq i \leq log_2 n$. The network thus obtained has a $log_2 n$ diameter.

We now define the Postage Stamp Problem and then relate it to circulant graphs. The Postage Stamp Problem [5] is defined as:

Postage Stamp Problem (PSP): Given k and h, find a set of k denominations $\{S_1, S_2, ..., S_k\}$ such that

(a) sums of h (or fewer) of these integers can realize the numbers $1, 2, 3, \ldots, N$

(b) the value of N in (a) is as large as possible.

To understand this we look at a small example where a post office issues stamps in sets of k denominations and allows customers to use no more than h stamps. They wish to choose for a given h and k, a set which gives them the greatest such N such that $1,\ 2, \ldots,\ N$ is possible to get using h (or fewer) of these k denominations. For example, if k = 4 and h = 3, this set is (1, 4, 7, 8) cents. Using 3 or fewer of these stamps, any value less than or equal to 24 cents can be achieved.

Stating the problem formally, let h, k be numbers (i.e non-negative integers). Let $S = \{\ S(1),\ S(2), ...,\ S(k)\}$ be a set of distinct numbers (stamp denominations) such that $S = \{\ S(1) <\ S(2) < \ldots <\ S(k)\}$, and a value *v* is obtainable from S if there exists another set C such that

$$v = \sum_{i=1}^{k} C(i)S(i) \text{ and } \sum_{i=1}^{k} C(i) \leq h.$$

V(k,h) for a set $S = \{S_1, S_2, ..., S_k\}$ is defined as the maximum value obtainable using h (or fewer) of S_i's. An optimal set $S^o = \{S_1^o,\ S_2^o, \ldots,\ S_k^o\}$ (of which there may be more than one) for a given k ,h is one whose value is the maximum = $N(k,\ h)$.

We are interested in $N(k,\ h)$ and the set S^o for such a $N(k,\ h)$. In the next section we relate the Postage Stamp Problem to circulant graphs.

2.1 Postage Stamp Problem in relation to Circulant Networks

We now relate Circulant Graphs and PSP mentioned in the previous section. Recall that s_i's give the neighbors of a node in circulant graphs and S_i^o's for a given h and k give us the optimal set S^o which can be used to get to a value from 1 to $N(k,\ h)$. Given a solution to PSP for a given h and k we can construct a circulant graph of size $N(k,\ h)$. The idea is to make the finger pointer offsets equal to the stamp denominations and the network diameter will then be equal to the number of stamps (i.e. h).

Formally, let $s_1 = S_1^o,\ s_2 = S_2^o, \ldots,\ s_k = S_k^o$. Let h_{ij} be the number of hops, chosen optimally (using shortest paths), required to reach node n_j from n_i. Any node n_j is less than $N(k,h)$ distance away from n_i in identifier space. Also, any number from $1, \ldots,\ N(k,h)$ can be reached using at most h of S_i^o's. Thus, $\forall(i,j) h_{ij} \leq h$.

In that case, for a given $(k,\ h)$, S^o provides us the optimal values of s_i's such that the graph constructed using these s_i's has the number of hops (chosen optimally) bounded by h.

Though the set S_i^o's provides us with an optimal set, it still does not tell us how to choose the next hop node for a key search. We would actually have to calculate the shortest paths in order to decide upon the next hop node for a given key search. The Postage Stamp Problem would give us the optimal $N(k,h)$ which no other circulant graph can better. We provide some results to the Postage Stamp Problem in the next section.

2.1.1 Existing Solutions to PSP

The Postage Stamp Problem has been studied widely. Inspite of many efforts to solve this problem, it still remains largely unsolved. Solutions to the Postage Stamp problem for some values of h and k have been evaluated. In this section we present these existing solutions to the Postage Stamp Problem. Solutions $N(k,\ h)$ to

the PSP would serve as an upper bound to our approach in the next section.

For k = 1, the only possible value of the set S^o = $\{1\}$. Thus, $N(1,\ h)\ =\ h\ \forall h \geq 1$. For k = 2, for any set S = $\{1,\ S_2\}$, $N(2,\ h)$[10] was calculated as:

$$N(2,\ h) = (h - S_2 + 3)S_2 - 2.$$

The maximum value of $N(2,\ h)$ is achieved for:

$$S_2^o = \begin{cases} 1/2(h+3) & \text{if h is odd,} \\ 1/2(h+3 \pm 1) & \text{otherwise.} \end{cases}$$

Therefore,

$$N(2,\ h) = \lfloor \tfrac{h^2+6h+1}{4} \rfloor,$$

and the corresponding set S^o = $\{1, S_2^o\}$. For $k = 3$ [3], the following optimal set S^o = $\{1,\ S_2^o,\ S_3^o\}$ for h sufficiently large was calculated:

$$S_2^o = 2\lfloor \tfrac{4h+4}{9} \rfloor - \lfloor \tfrac{2h}{9} \rfloor + 3,$$
$$S_3^o = (\lfloor \tfrac{2h}{9} \rfloor + 2)S_2 - \lfloor \tfrac{4h+4}{9} \rfloor - 2.$$

Using the above $N(3,\ h)$ is calculated as: $N(3,h)\ =\ (h\ -\ \lfloor \frac{4h+4}{9} \rfloor\ -\ \lfloor \frac{2h}{9} \rfloor)S_3^o\ +\ \lfloor \frac{2h}{9} \rfloor S_2^o\ +\ \lfloor \frac{4h+4}{9} \rfloor$.
Later, it was shown[4] that above equation holds for $h \geq 200$. It was then verified[6] for $23 \leq h \leq 200$ and the optimal set S^o and $N(k,\ h)$ for $1 \leq h \leq 22$ were calculated. In spite of the great effort of many researchers, there is still no exact formula available for k=4. We provide the set of known values in the Table 1. Due to lack of space we do not list all the known values of PSP.

The only nontrivial upper bound[8] available in the general case is given as:

$$N(k,\ h) \leq \tfrac{(k-1)^{k-2}}{(k-2)!}(\tfrac{h}{k})^k + O(h^{k-1}),$$

3 Largest Denomination First

We look at the slight variant of the Postage Stamp Problem called Largest Denomination First (LDF). We use S^l and $N^l(k,\ h)$ for the optimal set of denominations and the maximum value of N such that numbers from $1,\ 2, \ldots,\ N$ are achievable using h or fewer of these denominations. We now formally define Largest Denomination First.

Largest Denomination First: Given k, h find S^l = $\{S_1^l, S_2^l, \ldots, S_k^l\}$ such that:

(a) $N^l(k,\ h)$ is maximized and $1,\ 2, \ldots, N^l(k,\ h)$ are achievable using at most h denominations,

(b) assuming, to achieve a value of n you always use the largest denomination S_L first, such that $S_L \leq n$, update $n = n - S_L$, and repeat this until $n = 0$.

We now explain why the above mentioned problem would be interesting to solve. In Chord, the next hop neighbor is chosen as the node less than the target node but closest to it in the identifier space. This is the same as choosing the largest available denomination such that you do not overshoot n, where n would represent the distance in identifier space between the node in question and the target node. In this strategy we are guaranteed to make progress and get closer to the target. In the remainder of this section we present our algorithm to solve the above mentioned problem. The set S and the maximum achievable value N for our solution to the above problem are represented as S^t and $N^t(k,\ h)$ respectively.

Definition: $N^t(h,\ k,\ a)$ is the maximum value achievable using at most a denominations of the set S^t = $\{S_1^t, S_2^t, ..., S_k^t\}$.

Note that the set S^t is the same as in $N^t(k,\ h,\ h)$. Also, note that $N^t(k,\ h,\ h) = N^t(k,\ h)$. In our further discussion, we shall use $N^t(k,\ h,\ h)$ for consistency.

We use dynamic programming to provide an approximate solution to the above problem. Assuming that the solution for $\forall g \leq h,\ N^t(k - 1,\ g,\ g)$ has already been calculated, our problem now resides in calculating the k denominations S^t for $N^t(k,\ h,\ h)$.

h/k	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	8	12	16	20	26	32	40	46
3	3	7	15	24	36	52	70	93	121	154
4	4	10	26	44	70	108	162	228	310	
5	5	14	35	71	126	211	336	524		
6	6	18	52	114	216	388	638			
7	7	23	69	165	345	664	1137			
8	8	28	89	234	512	1045	2001			
9	9	34	112	326	797	1617				
10	10	40	146	427	1055	2510				

Table 1: Best known values for PSP

3.1 Calculating the k^{th} denomination

Let $D_i(k,\ h)$ be the choice of denomination S_k^t when at most $h-i$ stamps are to be used on the remaining $k-1$ denominations (i.e other than the denomination $D_i(k,\ h)$) and $N_i^t(k,\ h,\ h)$ be the corresponding maximum achievable value using this denomination as the k^{th} denomination with the remaining $k-1$ denominations being equal to the $k-1$ denominations for $N^t(k-1,\ h-i,\ h-i)$. In that case, the value of D_i would be chosen as one greater than the value achievable using $h-i$ stamps with one fewer denomination (i.e excluding the D_i denomination), since this is the minimum value not representable using $(h-i)$ stamps.

So,

$$D_i(k,\ h) = N^t(k-1,\ h-i,\ h-i) + 1 \quad (1)$$

We now calculate the value of $N_i^t(k,\ h,\ h)$. If we allocate i stamps to $D_i(k,\ h)$ then we are left with $(h-i)$ stamps. The maximum value that we can reach using $(h-i)$ stamps and not using the $D_i(k,\ h)^{th}$ denomination is

$$V = N^t(k-1,\ h-i,\ h-i)$$

Thus, we would get the achievable value as:

$i * D_i(k,\ h) + N^t(k-1,\ h-i,\ h-i)$

But V is one less than the value of $D_i(k,\ h)$. So, rather than using i stamps on $D_i(k,\ h)$ we can use $(i+1)$ stamps on $D_i(k,\ h)$ and the remaining $(h-i-1)$ on the other $(k-1)$ stamps. In that case the maximum value of $N_i^t(k,\ h,\ h)$ is given as:

$N_i^t(k,\ h,\ h) = (i+1) * D_i(k,\ h) + N^t(k-1,\ h-i,\ h-i-1)$

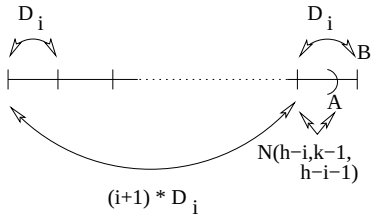

Figure 1: Value of $N^l(h,k)$

If we allocate more than $(i+1)$ stamps to D_i (say $i+2$) then as shown in Figure 1 the values from A=$(i+1) * D_i(k,\ h) + N^t(k-1,\ h-i,\ h-i-1)$ to B=$(i+2) * D_i(k,\ h)$ will not be representable using these denominations since

$N^t(k-1,\ h-i,\ h-i-1) < N^t(k-1,\ h-i,\ h-i) + 1$

The problem now remains to calculate the value of $N^t(h,k,h-1)$. When $h \geq k$ and we want to spend $h-1$ stamps on $\{S_1^t,\ S_2^t, \ldots,\ S_k^t\}$, we can take the solution for $N^t(k,\ h,\ h)$ and reduce the number of stamps that it allocates to S_k^t by 1. This reduction will not however change how we chose to allocate stamps among the remaining S_i^t's, so $N^t(k,\ h,\ h) - S_k^t$ will give us a suitable value for $N^t(k,\ h,\ h-1)$. However,

when $h < k$, we are not guaranteed that the allocation among the remaining S_i^t's would remain unchanged, so we use the allocation explicitly chosen for $k-1$ denominations.

The maximum value $N^t(h,k)$ over all such i is calculated as:

$$N^t(k,\ h,\ h) = max(N_i^t(k,\ h,\ h))\ 1 \leq i \leq h. \tag{2}$$

and the S_k^t'th denomination is calculated as: $S_k^t = D_i(k,h)$ such that $N_i^t(k,h,h) = N^t(k,h,h)$.

In the next section we present our results and show for which values of h and k our algorithm would be fairly optimal.

4 Results

In this section we present results to our algorithm stated in the previous section. Table 2 presents the value of N we are able to achieve using given values of h and k.

We further divide our discussion on values of h and k and then discuss the tradeoff solutions for constructing a network of given size N.

case: h=k

We observe that the values for $N^t(k,\ h)$ we achieve in this case, are equal to $F(2h+1)-1$ where $F(x)$ is the x'th Fibonacci sequence given as:

$$F(x) = \frac{(1+\sqrt{5})^x - (1-\sqrt{5})^x}{(\sqrt{5})2^x}$$

We have calculated the value of $N^t(k,\ h)$ for $h = k = i$, $1 \leq i \leq 50$ and have verified $N^t(k,\ h)$ to be equal to $F(2h+1)-1$. So, we conjecture that for $h = k$ and $h > 50$, N^t is given as $F(2h+1)-1$. For $h = k = i$, $1 \leq i \leq 50$ we provide the following comparisons to Chord.

For Chord the value of N^c is given as :

$$N^c(k,\ h) = 2^h$$

If we define the ratio N_r as N^t/N^c and substitute the corresponding values for N^t, N^c we get the following equation:

$$N^r = (1.144123)^{2h} * 0.7236 + (0.437016)^{2h} * 0.276$$

The ratio of the number of finger pointers for a particular value of N required by our algorithm (h^t) to that required by Chord (h^c) is calculated to be:

$$h^t \approx 0.72 h^c \tag{3}$$

The above equation shows that in comparison to Chord, we can reduce the size of the finger pointers as well as the size of the network diameter to 28%.

For h=15,k=15, we can construct a network of size 1,346,268 with corresponding S_i^t's as: 1, 2, 5, 13, 34, 89, 233, 610, 1597, 4181, 10946, 28657, 75025, 196418 and 514229. For h=20, k=20, we can construct a network of size 165,580,140, with corresponding S_i^t's as: 1, 2, 5, 13, 34, 89, 233, 610, 1597, 4181, 10946, 28657, 75025, 196418, 514229, 1346269, 3524578, 9227465, 24157817 and 63245986.

case: h > k

Let us assume we have an exponentially increasing denominations $S_1^t,\ S_2^t, \ldots,\ S_k^t$. In this case, the approach of taking the largest denomination first to reach a value of n would lead us to the optimal number of hops (stamps). In our case, we select the denomination S_k^t as the largest number N_j that cannot be achieved using $k-1$ denominations, $S_1^t,\ S_2^t, \ldots,\ S_{k-1}^t$. This is because all the values before these are achievable using the $k-1$ denominations. If we have enough stamps to use S_{k-1} as one of the stamps at least once, in achieving N_j, then the value of S_k will be much greater than the value of S_{k-1} providing an exponential growth in the size of the new denominations. So, in that case our approximation would work quite well.

case: h < k

When $h < k$, our assumption that a solution for $k-1$ denominations is a subproblem for k denominations no longer holds true, so our algorithm does not find good denominations. In reviewing the results of brute-force calculation of

h/k	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	7	10	13	16	19	22	25	28
3	3	7	12	20	28	36	44	52	60	68
4	4	10	20	33	54	75	96	117	138	159
5	5	14	29	54	88	143	198	253	308	363
6	6	18	40	78	143	232	376	520	664	808
7	7	23	55	111	206	376	609	986	1363	1740
8	8	28	71	152	294	541	986	1596	2583	3570
9	9	34	90	208	417	772	1418	2583	4180	6764
10	10	40	114	268	570	1100	2023	3714	6764	10945

Table 2: Results for Largest Denomination First using our algorithm.

optimal denomination sets, what we find is that the best denomination sets for $h << k$ incorporate something close to base-b arithmetic, where $b = \frac{k}{h}+1$. For $(k = 18, h = 2)$, for example, the denominations of $1, 2, ..., 9, 10, 20, ..., 90$ have a maximum sum of 100 using 2 stamps. Similarly, for $(k = 27, h = 3)$, denominations of $1, 2, \ldots, 9, 10, 20, \ldots, 90, 100, 200, \ldots, 900$ seem to work well and allow sums up to 1010. We are actively pursuing an alternative algorithm that can find optimal sets of denominations under such scenarios.

4.1 Tradeoff Utilization

We now illustrate how our algorithm allows us to benefit from the tradeoff that exists between the finger pointers and the network diameter. We are able to construct a network of a given size n using various values of h, k. So, if we want to construct a graph of size one million, we could do it in different ways. Here we mention three possible ways of doing this in our construction, (h=15, k=15), (h=12, k=19) and (h=20, k=13).

5 Conclusion and Future Work

We provide an efficient way of constructing ring-like structures for a given network diameter and number of finger pointers using the Postage Stamp Problem. We also provide an upper bound on the size of the circulant graphs that can be constructed using a given h and k. Moreover, we provide tradeoff constructions using our approach so that a given value of n can be achieved using various values of h, k. Additionally, with a routing table of size 20 and a network diameter of 20, we can construct a network of 165 million nodes in comparison to Chord's 1 million nodes.

Our constructions so far address the case when the number of finger pointers is comparable to the number of hops. In case of many finger pointers and few hops ($h < k$), we are actively pursuing research into better construction algorithms.

References

[1] J. C. Bermond, F. Comellas, and D. F. Hsu. Distributed loop computer networks: A survey. *Journal of Parallel and Distributed Computing*, 24(1):2–10, 1995.

[2] Krishna Gummadi, Ramakrishna Gummadi, Steven Gribble, and Henry Levy. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of the 2003 ACM SIGCOMM Conference*, 2003.

[3] G. Hofmeister. Asympotische abshatzungen für dreielementige extermalbasen in

naturlichen zahlen. *J. Rein Angew. Math.*, 232:77–101, 1968.

[4] G. Hofmeister. Die dreielementigen extremalbasen. *J. Rein Angew. Math.*, 339:207–214, 1983.

[5] W.F. Lunnon. A postage stamp problem. In *The Computer Journal*, pages 377–380, 1969.

[6] S. Mossige. On the extremal h-range of the postage stamp problem. *Math. Comput.*, 36:575–582, 1981.

[7] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press, 2001.

[8] Ö. J. Rödseth. An upper bound for the h-range of the postage stamp problem. *Acta Arithmetica*, 54(4):301–306, 1990.

[9] Antony Rowstron and Peter Druschel. Past: Persistent and anonymous storage in a peer to peer networking environment. In *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems*, pages 65–70, 2001.

[10] A. Stöhr. Deloste und ungelöoste fragen über basen der naturlichen zahlenreihe,i, ii. *J. Rein Angew. Math.*, 194:40–65, 1955.

[11] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.