CDACAN: A Scalable Structured P2P Network Based on Continuous Discrete Approach and CAN

Lingwei Li[1], Qunwei Xue[2], and Deke Guo[1]

[1] School of Information System and Management
National University of Defense Technology, Changsha, Hu Nan, 410073, P.R. China
[2] China Institute of Geo-Environment Monitoring, Beijing 100081, P.R. China
lilingwei_82104@126.com, qwxue@hotmail.com, guodeke@gmail.com

Abstract. CAN is a famous structured peer-to-peer network based on d-dimensional torus topology with constant degree and logarithmical diameter, but suffers from poor scalability when $N \gg 2^d$, N is the number of peers. To address this issue, we proposes a novel scalable structured peer-to-peer overlay network, CDACAN that embeds the one-dimensional discrete distance halving graph into each dimension of CAN. The out-degree and average routing path length of CDACAN are $O(d)$ and $O(log\ (N))$, respectively, and are better than that of CAN. On the other hand, we analyze the optimal value of dimensions and the smooth division method of d-dimensional Cartesian coordinate space when handling the dynamic operations of peers. The smooth division of multidimensional space can improve the routing performance, and also is helpful to keep load balance among peers. Those properties and protocols are carefully evaluated by formal proofs or simulations. Furthermore, we present a layered improving scheme to decrease the out-degree of each peer in the future work. The expected topology will keep 8 out-degree and $O(log_2(N)+d)$ routing path length.

1 Introduction

Structured peer-to-peer networks, abbreviated as P2P, have been proposed as a good candidate infrastructure for building large scale and robust network applications, in which participating peers share resources as equals. They impose a certain structure on the overlay network and control the placement of data, and thus exhibit several unique properties that unstructured P2P systems lack.

Several interconnection networks have been used as the desired topologies of P2P networks. Among existing P2P networks, CAN [1] is based on the d-dimensional torus topology; Chord[2] Pastry[3] and Kademlia [4] are based on the hypercube topology; Viceroy[5] is based on the butterfly topology; Koorde [6], Distance Halving network [7], D2B [8], ODRI [9] and Broose [10] are based on the de Bruijn topology; FissionE[11], MOORE[12] are based on the Kautz topology.

CAN is a famous structured peer-to-peer network based on d-dimensional torus topology with constant degree and logarithmical diameter. It supports multi-dimensional exact and range queries in a natural way, but suffers from poor scalability when $N \gg 2^d$, N is the number of peers. To address this issue, we introduce

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 611–622, 2007.

CDACAN, an improved structured P2P network based on CAN and the CDA (Continuous-Discrete Approach). CDACAN is a constant degree DHT with $O(log(N))$ lookup path length and inherits the properties of Cartesian coordinate space. The main contributions of this paper are as follows:

- We extend one-dimensional discrete distance halving graph to multi-dimensional space, abbreviated as *DGd*, to serve as the topology of CDACAN. The *DGd* can keep the good properties of Cartesian coordinate space.
- We improve the routing algorithms proposed in literature [7] to support *DGd* . Thus, CDACAN achieves a less value of the average routing path length than that of the traditional CAN.
- We design algorithms to calculate the necessary precision of destination code to perform the search. We reveal that the smoothness and regularity of the spatial division can reduce the hazard in the estimation of precision and neighbor forwarding can amend the routing.
- We design the peer joining and departing protocols based on an associate sampling mechanism to improve the smoothness and regularity of all cells of a whole Cartesian coordinate space after division.
- We point out that the out-degree of each peer in CDACAN is still relative high, and propose a layered scheme to reduce the value of out-degree. It achives a better tradeoff between the routing table size and the routing delay just as the Cycloid does.

The rest of this paper is organized as follows. Section 2 introduces some useful conceptions in CDA. Section 3 proposes the topology construction mechanism, the routing algorithm, and the joining, departing protocols for CDACAN. Section 4 evaluates the characteristics of CDACAN by comprehensive simulations. Conclusion and future work are discussed in Section 5.

2 Basic Concepts of Continuous Discrete Approach

Continuous Distance Halving Graph *Gc***:** The vertex set of *Gc* is the interval I= [0, 1). The edge set of *Gc* is defined by the following functions: $L(y)=y/2$, $R(y)=y/2+1/2$, $B(y)=2y \ mod\ 1$, where $y \in I$. L abbreviates 'left', R abbreviates 'right' and B abbreviates 'backwards'.

Discrete Distance Halving Graph *Gd***:** X denotes a set of n points in I. A point X_i may be the ID of a processor V_i or a hashing result of the value of ID. The points of X divide I into n segments. Let us define the segment of X_i to be $S(X_i) = [X_i, X_{i+1})$, i=1,2,…n−1, and $S(X_n)=[X_n, 1) \cup [0, X_1)$. Each processor V_i is associated with the segment $S(X_i)$. If a point y is in $S(X_i)$, we say that V_i covers y. A pair of vertices (V_i, V_j) is an edge of *Gd* if there exists an edge (y, z) in the continuous graph, such that $y \in S(X_i)$ and $z \in S(X_j)$. The edges (V_i, V_{i+1}) and (V_n, V_1) are added such that *Gd* contains a ring. In figure 1, the upper diagram illustrates the edges of a point in *Gc* and the lower diagram shows the mapped-intervals of an interval that belongs to *Gd*.

The greedy routing and distance halving routing algorithms apply to the graph, and are abbreviated as GA and DHA, respectively. If the latter adopts the binary string representation of the destination X_d to perform the first step, we call it as direct distance halving algorithm, abbreviated as DDHA. We also use continuous discrete routing algorithm, abbreviated as CDRA, as a general name for both GA and DDHA.

3 CDACAN

In order to improve the scalability of CAN, we embed a d-dimensional discrete distance halving graph into CAN, and achieve a novel P2P network.

3.1 Topology Construction Mechanism

We extend the conceptions of *Gc* and *Gd* to multidimensional space.

d*-Dimensional Continuous Distance Halving Graph** *DGc:** The vertex set of *DGc* is the unit *d*-dimensional Cartesian coordinate space $S=\{(X_0,X_1,\ldots\ldots X_{d-1})|X_i \in [0,1), i=0,1,\ldots\ldots d-1\}$. Given $Y=(Y_0, Y_1,\ldots\ldots Y_{d-1})$, the edge set of *DGc* is defined by the following functions: $DL(Y)=\{(X_0,X_1,\ldots X_j,\ldots X_{d-1})|X_i=Y_i,\ i=0,1,\ldots j-1,\ j+1\ldots d-1, X_j=Y_j/2,\ j=0,1,\ldots\ldots d-1\ \}$; $DR(Y)=\{(X_0,X_1,\ldots X_j,\ldots X_{d-1})|X_i=Y_i,\ i=0,1,\ldots j\text{-}1,\ j+1\ldots d-1, X_j=Y_j/2+0.5,\ j=0,1,\ldots\ldots d-1\ \}$; $DB(Y)=\{(X_0,X_1,\ldots X_j,\ldots X_{d-1})|X_i=Y_i,\ i=0,1,\ldots j-1, j+1\ldots d-1, X_j=2Y_j \bmod 1, j=0,1,\ldots\ldots d-1\ \}$;

d*-Dimensional Discrete Distance Halving Graph** *DGd:** *P* denotes a set of *N* processors, divide *S* into *N* zones and each zone Z_j is charged by P_j in *P*, $j=1,2,3,\ldots N$. A pair of vertices (P_i, P_k) is an edge of *DGd* if there exists an edge (y, z) in *DGc*, such that $y \in Z_i$ and $z \in Z_k$. The CAN-like edges are included in *DGd* too.

Let *Z* denote the zone set of *DGd*, Z_i is a zone that $Z_i \in Z$ and has a serial number of i, $i=1,2,3\ldots N$. $Z_i.j$ denotes the interval of the *j*th dimension of Z_i, $j=0,1,2,\ldots d-1$.

CLAIM1: let N_i is a node in *DGd*, whose zone is Z_i with volume *V*. Z_i is mapped to *d L*-mapped zones, *d R*-mapped zones and *d B*-mapped zones, and their volumes are $V/2$, $V/2$ and $2V$ respective.

The definitions, corollaries and lemmas mentioned below will be used in later.

Definition 1. The **length** of a binary decimal *a* is $\boldsymbol{L(a)}$, where $L(a)$ is the number of bits behind the radix point of *a*. If $L(a)=i$, *a* is a ***i*-regular-point**.

Definition 2. Let an interval $I=[a, b)$, where *a* is a binary decimal and $L(a)\leq i$, $b=a+2^{-i}$. We call it ***i*-regular-interval**. Both *a* and *b* are the **boundaries** of *I*.

We can infer a corollary and two lemmas on the base of *definition 2*. The related proofs are omitted due to the page limitation.

Corollary 1. In *Gd*, the mapped intervals of *i*-regular-intervals induced by $R(y)$ or $L(y)$ are $(i+1)$-regular-intervals; and that induced by $B(y)$ are $(i-1)$-regular-intervals.

Lemma 1. Any *i*-regular-interval $I_a=[a, b)$ possesses a unique binary prefix *Sp* whose length is *i*. Every point in [0,1) whose binary form with prefix *Sp* is included in I_a; and the *i*-prefix of every point *p* in (a, b) is *Sp* and *p* satisfies $L(p)>i$.

Lemma 2. There must be only one boundary point *c* of *i*-regular-interval *I* satisfies: $L(c) = i$, the length of other boundary point *e* satisfies $L(g)< i$.

Definition 3. The **smoothness of *i*th dimension** and **smoothness of volume** are denoted as $SM_i(Z)$ and $SM_V(Z)$, respectively. They are defined to be the existing number p that satisfies $MAX_{j,k}$ $|Z_j.i|/|Z_k.i|<p$, $i = 0,1,2,\ldots\ldots d-1$, and $MAX_{j,k}$ $|Z_j|/|Z_k|<p$ for any network size N, respectively.

Definition 4. The **regularity of *Z*** means that all the $Z_i.j$ are regular-intervals, i=1,2,3…N, j=0,1,2,…d−1.

The smoothness problem has been discussed in [7]. Its effect on routing algorithms, however, is still an open problem. We show that CDA can generate a series of regular intervals, and its essential objective is to gain the unique prefix of the interval (zone).

Theoretic Analysis of Degree distribution: supposing the spatial division is **smooth and regular**. For d-CDACAN, the degree of a node N_i is about 6d, the edges including: about 2d CAN-like edges to the neighbor of Cartesian coordinate space; about d, d and 2d edges induced by *DR*, *DL* and *DB* functions in *DGd* respectively.

Figure 2 shows a 2-*DGd*, the overstriking area is an actual zone Z_i whose owner is N_i; red areas are *R*-mapped zones and *L*-mapped zones of Z_i; green areas are *B*-mapped zones of Z_i. The number of degree of N_i is 16, including CAN-like edges.

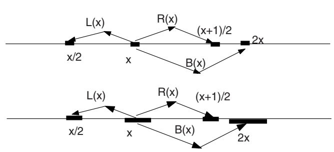

Fig. 1. node degree in *Gc* and *Gd*

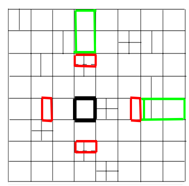

Fig. 2. A possible 2-DGd

It is hard to get the upper bound (mentioned in *Theorem1* in [7]) of the number of edges of any node in *DGd*. And it is clear that the spatial division of high dimensional *DGd* without smooth volumes and regular shapes may lead to a high value of degree.

3.2 Routing Protocol

3.2.1 Routing Algorithm

Let X_d denote a lookup destination, $I_d = [I_x, I_y)$ is the interval that contains X_d and V_d is the processor that owns I_d. I_s is the interval of the lookup initiator V_i. I_r is the longest regular-interval in I_d and $L(I_r)=h$, and I_L is the longest regular-interval in I_s and $L(I_s)=v$. Our discussion in 3.2.1 and 3.2.2 is general, and does **not base on the hypothesis of smoothness and regularity**. We will reveal three problems that have not been pointed out and clarified in [7].

- In *DGd*, how does CDRA calculate I_L and I_r?
- They did not reveal why the CDRA is prone to fail in distributed scenario, and how the smoothness and regularity of spatial division reduce the hazard of routing, and how neighbor forwarding amends the routing.
- They did not reveal the key factors that affect the routing path length in CDRA.

The three problems are interconnected and will be settled later.

The routing algorithm of CDACAN is **routing on every dimension obeyed to the improved CDRA respectively**. The improved CDRA include three steps: calculating appropriate I_r (for both GA and DDHA) and I_L (for GA only); routing by CDRA; neighbor forwarding (if necessary). We will introduce our contributions, which are the first and third steps.

Because the improved CDRA is performed on every dimension of CDACAN, we adopt *Gd* as the background of improved CDRA for convenience and the adjective of "improved" may be omitted in the latter.

3.2.2 Calculation of I_r (I_L) Under the Global Division Information

We assume that the interval division states are known by all the peers in this section and devise an algorithm to calculate I_r (I_L) under the hypothesis. We will introduce a lemma first.

Lemma 3. Let x and y denote two binary decimals and $1>y>x\bullet 0$, there must exist one binary decimal m in $I = [x, y)$ whose length $L(m)$ is the smallest, We call m the **Shortest Normative Point**.

Then we will propose an algorithm to calculate m. The functions of binary string operations in the pseudo code are quite straightforward.

Algorithm 1. FindShortestBinaryDecimal(*x*, *y*)

```
Require: The inputs must satisfy that x< y
t • y − x
z • t.PreserveLargest1Bit()
L • z.GetLength()
if y = y.PreserveBits(L) then
    m • y − 2^-L
else
    m • y.PreserveBits(L)
p • m − 2^-L
if p • x and m.GetLength()>p.GetLength() then
    m • p
return m
```

Corollary 2. The Shortest Normative Point of a regular interval $I=[a, b)$ is a.

Theorem 1. I_r can be calculated by the algorithms mentioned as follows.

Algorithm 2. DetermineRoutingLengthLowerBound (I_d)

```
Require: I_d=[I_x, I_y) what has been defined in 3.2.1.
t • I_y − I_x
z • t.PreserveLargest1Bit()
h • z.GetLength()
StartPoint • FindShortestBinaryDecimal(I_x, I_y)
I_r1 = SearchLeft(StartPoint, h, I_x)
I_r2 = SearchRight(StartPoint, h, I_y)
return I_r = max(I_r1, I_r2)
```

```
Algorithm 3. SearchLeft(StartPoint, h, I_x)
Require: StartPoint is a point in I_d, and h is a positive integer.
if (StartPoint - 2^-h) < I_x) then
        I_r1= SearchLeft(StartPoint, h + 1, I_x)
else
        I_r1=[StartPoint - 2^-h, StartPoint)
return I_r1
```

```
Algorithm 4. SearchRight(StartPoint, h, I_y)
Require: StartPoint is a point in I, and h is a positive integer.
if (StartPoint + 2^-h) • I_y) then
        I_r2= SearchRight (StartPoint, h+1, I_y)
else
        I_r2= [StartPoint, StartPoint + 2^-h)
return I_r2
```

Proof. let the Shortest Normative Point of I_d =[I_x, I_y)is m and $I_1 = m-I_x$, $I_2 = I_y-m$. We can decompose I_1 and I_2 into the formulas whose form is $x_1 2^{-1}+ x_2 2^{-2}+\ldots x_i 2^{-i}+\ldots$, x_i = 0 or 1, i=1,2,……. If $L(m)=L$ there must be:

$$I_1 = x_1 2^{-L}+x_2 2^{-L-1}+\ldots x_{i+1-L}\, 2^{-i}+\ldots,\ x_i = 0 \text{ or } 1,\ i=L, L+1\ldots\ldots \tag{1}$$

$$I_2 = y_1 2^{-L}+y_2 2^{-L-1}+\ldots y_{i+1-L} 2^{-i}+\ldots,\ y_i = 0 \text{ or } 1,\ i=L, L+1\ldots\ldots \tag{2}$$

The relationship between the decomposition of I_1 and *Algorithm3* is that *Algorithm3* can find the largest regular-interval I_{r1} at the left of m, corresponding to the first number appeared in (1).

And *SearchRight* can find a regular-interval I_{r2} at the right of m too. □

It is obvious that DDHA will adopt the unique prefix of I_r to perform the routing when the interval division state is known by all peers. GA can get I_r and I_L by the same method too. The detailed explanation of DDHA and GA can be found in [7].

Corollary 3. if I_d and I_s are regular-intervals, then $I_r = I_d$, $I_L = I_s$, respectively.

3.2.3 Neighbor Forwarding and the Effects of Smoothness and Regularity

The Algorithm in 3.2.2 bases on a hypothesis that the interval division is known by all peers. The hypothesis can not be satisfied because of the distributed property of P2P network. If V_i wants to query a resource whose ID is X_d. Firstly V_i have to estimate I_d what contains X_d to perform *Algorithm2*, the estimation including the length and the beginning point of I_d, and the estimation of I_d is denoted by I_d'; Secondly V_i adopt *algorithm2* to calculate I_r' and I_L by I_d' and I_s respectively; Thirdly V_i perform DDHA by I_r' or GA by I_r' and I_L respectively; Fourthly, the searching massage is sent to a processor denotes by V_d'; At last, V_d' is not always V_d because of the uncertainty of the estimation of I_d, and the method for amending is **neighbor forwarding**.

Let suppose L_{ave} is the average interval length in the network. The effect of the smoothness is making the calculation of L_{ave} accurately, thus $L(I_d')$ too; and the effect of regularity is making the position of the entire intervals regular. In CDACAN, the regularity is achieved by the CAN-like divisional scheme, and the smoothness is achieved by the sampling operation in the joining and departing stages.

However, smoothness problem can not be addressed by a determinate method. So V_d' may be not V_d and we design a **neighbor forwarding** scheme for V_d', i.e. **forwarding a message to Cartesian-coordinate-neighboring nodes**, to amend the routing failure of CDRA what is resulted from a low-estimated I_r'.

3.2.4 Routing Path Length of Continuous Discrete Routing Algorithm

We can infer three corollaries when consider the problem under a **distributed scenario**.

Corollary 4. The key factor that affects the routing path length of GA is v.

Proof. In GA, V_i can calculate logical beginning point P_s as follows: gets the prefix of P_s by the unique prefix of I_L, and then gets the suffix of P_s by the unique prefix of I_r'. Because $P_s \in I_S$, the routing path length of original GA which starts at P_s is v. And I_r' affect the size of the routed message. □

Corollary 5. The key factor that affects the routing path length of DDHA is $h' = L(I_r')$.

Corollary 6. If the spatial division is smooth and regular, then $| h' - v | < log_2(p)$.

Proof. According to *Corollary3,* if the division is regular, there must be $I_r' = I_d'$ and $I_L = I_s$, and $|I_d'| = L_{ave} = 2^{-h'}$, $|I_s| = 2^{-v}$. According to the definition of smooth we know that existing a number $p>1$ what satisfies $1/p < |I_d'|/|I_s| < p$. So $|h' - v| < log_2(p)$. □

If the spatial division is smooth and regular, *Corollary6* implies that the routing path lengths of DDHA and GA are close. Else, it is difficult for V_i to determine appropriate I_d' to perform CDRA because V_i doesn't know I_d in distributed circumstance. GA can adopt a little larger I_d' to increase the probability of successful lookup and avoid neighbor forwarding because it does not waste too much bandwidth according to *Corollary4*; DDHA can not adopt too large I_d' because it increases the entire routing path length unnecessarily according to *Corollary5*. So the risk what comes from the uncertainty of the calculation of I_d' can do more harm to DDHA than GA.

Theoretic Analysis of Routing path length distribution: In smooth and regular division, the average routing path length and the diameter of DDHA and GA are both about L_{ave}, and there may be some lengths larger than L_{ave} because of neighbor forwarding caused by the impossibility of even spatial division. The hop number of neighbor forwarding is determined by smoothness.

3.3 Node Joins

As for most P2P networks, we assume there are some existing nodes as entry points of CDACAN, which can receive and process the node joining message. The joining procedure includes three stages: selects a zone to divide, redistributes resources, and updates routing tables. We will introduce the first step because it is unacquainted

The process is inspired by the join algorithm of *Gd* in [7]:

1. Estimate log_2N. The method to estimate log_2N is simple and is omitted here.
2. Sample $tlog_2N$ random points in d-dimensional unit space S, t is a constant and can be known by *Theorem11* [7], we also can select an appropriate t by simulation.

3. Check all zones that contain those points. Let N_l denote the node that contains the largest zone Z_l. Then divides Z_l into two parts in a similar way used by CAN.

Borrowing the *Theorem11* from *Gd* in [7], we can deduce the theorem as follow under the background of *DGd*.

Theorem 2. For any initial state, the largest zone would be of volume at most $1/2n$ after inserting n points.

The proof of theorem2 is omitted because it can be transplanted from [7] easily. This theorem provides the basis to keep smooth in *DGd* by the original sample scheme. We have improved the sample scheme by using the **local search** rule. It can reduce the number of messages used to carry out the sampling process and keep smooth.

Samples with Local Search: The sampled nodes check all the neighbors and select the neighbor whose zone is the largest. It can improve the efficiency and reduce the number of sampled points remarkably at the same time.

3.4 Node Departs

Because of the dynamic nature of P2P network, CDACAN need to consider not only node joining operation but also node departure operation. Updating the routing table and the reassigning zones are omitted because they are familiar. The departing protocol includes four steps as follows:

1. If N_D, whose zone is Z_D, wants to depart immediately, it selects the smallest neighbor node N_S to take over Z_D. So N_S manages two zones temporarily.
2. N_S samples number of $tlog_2N$ points in d-dimensional unit space S and send $tlog_2N$ messages to their owners, where the value of t is assigned in a same method used by the peer joining algorithm.
3. Every sampled node P_s searches "partition tree" to find a "leaf nodes" N_L by a similar algorithm used by the topology maintenance mechanism in CAN. Then P_s sends N_L back to N_S.
4. N_S selects the smallest leaf node N_{Le}, and then N_{Le} hand its zone Z_{Le} to its "geminate node" to merge to a valid zone and take over Z_D.

The first step ensures that the departure operation can be finished successfully. The last two steps can keep the regularity of the division for the same reason in CAN.

Our major contribution is the step two. The sample-based selection can improve the smoothness and does not bring too much sampling messages. By comparing with CDACAN, CAN adopt a simple scheme to find N_S and it can not deal with mass adversarial departure; the method in CDA needs to keep additional structure (bucket or Cyclic Scheme).

4 Evaluation

4.1 Query Path Length

In this section, we will verify the theoretical analysis results by comprehensive simulations. The query path length is measured in terms of overlay network hops. We

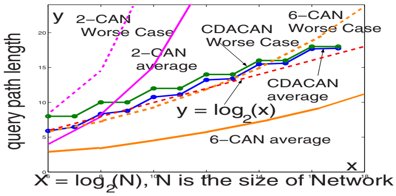

Fig. 3. Routing path length of CDACAN

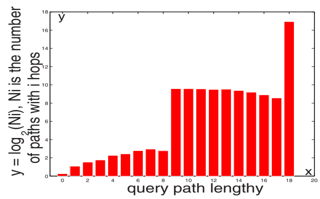

Fig. 4. Query path length distribution of CDACAN with 128k nodes

assume a simple model for the offered query load: queries are generated randomly and uniformly at each node, the desired query results are uniformly distributed in the key-space.

We plot the routing path length of 2-CDACAN in figure 3, and adopt 2-CAN and 6-CAN as the benchmark for our experiments since 2-CDACAN is based on 2-CAN and has the same degree with 6-CAN. We have simulated the worst-case and average-case path lengths as a function of the number of nodes N (x-axis, in the log scale). Figure 3 demonstrates that the routing path length of 2-CDACAN is close to the $y=log_2N$ function. The "step-like-increase" on both CDACAN routing path curves is due to the increasing of the estimated system size when N becomes larger. As expected, curves for the worse-case and average-case of 2-CAN and 6-CAN are coincide with that of functions $n^{1/2}$, $(1/2)n^{1/2}$, $3n^{1/6}$ and $(3/2)n^{1/6}$ respective. We also discover that 2-CDACAN is better than 2-CAN and inferior to 6-CAN. The simulation results also expose that the degree of each peer in CDACAN is relative high, and should decrease as less as possible. Figure 4 shows the distribution of path lengths in CDACAN with 128k nodes. We can observe that most paths with the length of $1+log_2(N)$. It is because our simulations adopt higher precision of I_r than the normal one for one bit to avoid neighbor search.

4.2 Sample in the Join Operation

The smoothness and degree problems of CDACAN have impact on the sample algorithm. We will analyze it by simulations.

We choose average degree to reveal the effects of local search as shown in figure 5. The upper line and lower line represent the average degree of peers in CDACAN that

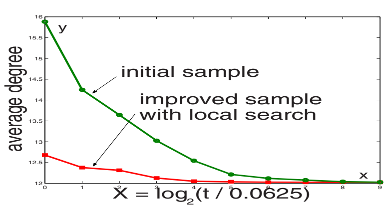

Fig. 5. Average degree of CDACAN with 1024 nodes

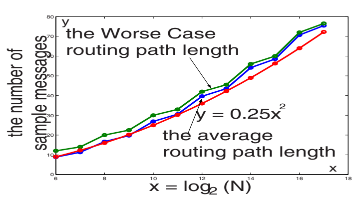

Fig. 6. The number of sample messages in CDACAN with $t = 0.25$

is generated by the original sampling and improved sampling with local search, respectively. We discover that the local search can improve the smoothness (Figure 9 and Figure 10) and reduce the degree (Figure 5) obviously.

Figure 6 illustrates that the average number of sample messages in a join operation increases logarithmically with respect to the size N. The number is the product of the routing path length and amount of samples. This experiment adopts t=0.25, which is a relative large value. The green line assumes the worse case of routing path and the blue line adopts the average. They are both close to y=0.25x^2 because the routing path length is close to $log_2(N)$ as shown in figure 3. The average number of overlay messages what are generated by the sample operation is less than 80 in CDACAN with 128k nodes. It illuminates that sample-based scheme in CDACAN is practical.

4.3 Volume and Smoothness

Volume distribution is an important network trait in CDACAN-like protocols what based on spatial division and spatial mapping. Figure 7 and Figure 8 demonstrate the effect of regular division. Figure 7 shows that irregular division leads to trivial distribution of volume and the ratio of the biggest zone and the smallest zone is not less than 20; Figure 8 shows that the regular scheme generates smooth division and the ratio of the biggest volume and the smallest volume is 4.

Figure 9 and Figure 10 demonstrate that the volume distribution of CDACAN becomes smooth when t increases. And they illustrate the volume distribution what is generated by the sample with local search and without local search respectively. Comparing figure 9 with figure 10, we discover that local search can improve the smoothness remarkably. And they uncover the causation of figure 5 too.

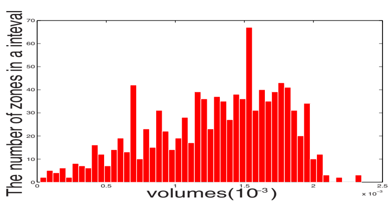

Fig. 7. Volume Distribution of CDACAN with irregular division

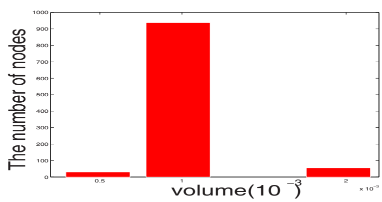

Fig. 8. Volume Distribution of CDACAN with regular division

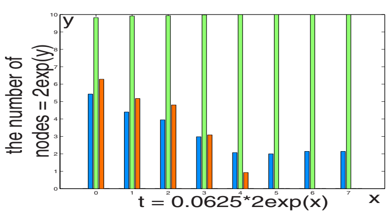

Fig. 9. The volume distribution in CDACAN without local search

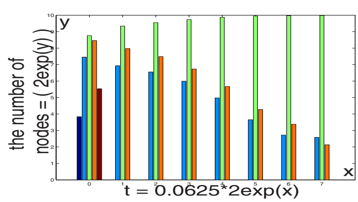

Fig. 10. The volume distribution in CDACAN with local search

4.4 Distribution of Degree

The distribution of degree in CDACAN is an important trait that represents the routing table overhead. The experiment which corresponds to Figure 11 is called E1, and the experiments which correspond to the green, blue and red bars in figure 12 are called E2, E3 and E4 respectively.

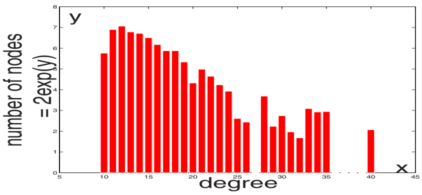

Fig. 11. Degree distribution in 1024 CDACAN by *t*=0.0625 without local search

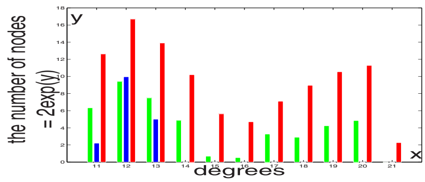

Fig. 12. Comparison of degree distribution

Figure 11 and the green and blue bars in figure 12 illustrate the degree distribution of experiments what have 1024 nodes. E1 adopts *t*=0.0625 without local search, and its volume distribution is shown with the first bar cluster in figure 9. E2 adopt *t*=0.0625 with local search; E3 adopt *t*=2 with local search. E2 and E3 correspond to the first and sixth bar cluster in figure 10 respectively. We can discover that **the distribution of degree is determined by the distribution of volume**. And the degree distribution is discrete because the volume distribution is discrete too.

Figure 5 shows that the average degree decreases when *t* increases. It is because the volume distribution becomes smooth as shown in figure 9 and figure 10. And the average degree of 2-CDACAN is about 12 what accords with the theoretical analysis. E4 adopt *t*=0.0625 with local search and 128k nodes. The setting of E4 and E2 is the same except for the network size. And we discover that their degree distributions are nearly the same. By reverse reasoning, we discover that their volume distributions are similar and CDACAN keep smoothness well under large network size.

5 Conclusions and Future Work

In this paper we have presented CDACAN, a scalable structured P2P network based on Continuous Discrete Approach and CAN. We firstly introduce the related conceptions in CDA. Then we introduce the design of our proposal detailedly: presenting the topology construction and the routing algorithm; In the routing algorithm, we propose algorithm to calculate the necessary precision of the destination, devise neighbor forwarding scheme to deal with the uncertainty of the routing in distributed circumstance, analyze the key factors that affect the routing path length of CDRA; introducing the join and departure protocols of CDACAN. At last we verify our protocols by simulation.

The main problem of CDACAN is its high degree. In CDACAN, the average routing path length and degree are about $log_2(N)$ and $6d$ respectively. We discover that the high degree can not reduce the path length. Our anticipant solution is layered improvement, named LCDACAN, whose **scheme is dividing *d*-CDACAN into *d* layers and the nodes in *i*-layer keep the edges on the *i*th dimension, *i*=0,1,…*d*−1**. The topology has $O(log_2(N)+d)$ routing path length and about 8 degrees on every node. LCDACAN keeps Cartesian coordinate space property and its tradeoff between routing path length and size of routing table is close to Cycloid [13].

References

1. Ratnasamy, S., Francis, P., Handley, M., Karp, R.M., Shenker, S.: A scalable content-addressable network. In: Proc. ACM SIGCOMM, San Diego, CA, pp. 161–172. ACM Press, New York (2001)
2. Stoica, I., Morris, R., Karger, D., Kaashoek, M., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. IEEE/ACM Trans. Networking 11(1), 17–32 (2003)
3. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
4. Maymounkov, P., Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric. In: Proc. International Peer-to-Peer Symposium, Cambridge, MA, USA, pp. 53–65 (2002)
5. Malkhi, D., Naor, M., Ratajczak, D.: Viceroy: A scalable and dynamic emulation of the butterfly. In: Proc. the 21st ACM PODC, Monterey,CA, pp. 183–192. ACM Press, New York (2002)
6. Kaashoek, F., Karger, D.: Koorde: A simple degreeoptimal distributed hash table. In: Proc. International Peer-to-Peer Symposium, Berkeley,CA,USA, pp. 98–107 (2003)
7. Naor, M., Wieder, U.: Novel architecture for p2p applications: the continuous-discrete approach. In: Proc. ACM Symposiumon Parallel Algorithms and Architectures, San Diego, California, USA, pp. 50–59. ACM Press, New York (2003)
8. Fraigniaud, P., Gauron, P.: D2B: a de bruijn based content-addressable network. Theor. Comput. Sci. 355(1), 65–79 (2006)
9. Loguinov, D., Casas, J., Wang, X.: Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. IEEE/ACM Trans. Networking 13(5), 1107–1120 (2005)
10. Gai, A., viennot, L.: Broose: a practical distributed hashtable based on the de bruijn topology. In: Proc. the International Conference on Peer-to-Peer Computing, Switzerland, pp. 167–174 (2004)
11. Li, D., Lu, X., Wu, J.: Fissione: A scalable constant degree and low congestion dht scheme based on kautz graphs. In: Proc. IEEE INFOCOM, Miami,Florida,USA, pp. 1677–1688. IEEE Computer Society Press, Los Alamitos (2005)
12. Guo, D., Wu, J., Chen, H., Luo, X.: Moore: An extendable peer-to-peer network based on incomplete kautz digraph with constant degree. In: Proc. 26th IEEE INFOCOM, IEEE Computer Society Press, Los Alamitos (2007)
13. Shen, H., Xu, Ch., Chen, G.: Cycloid: A constant-degree and lookup-efficient P2P overlay network. J. Performance Evaluation 63, 195–216 (2006)