Moore: An Extendable Peer-to-Peer Network Based on Incomplete Kautz Digraph With Constant Degree

Deke Guo*, Jie Wu†, Honghui Chen*, and Xueshan Luo*
* Key laboratory of C^4ISR Technology, College of Information Systems and Management
National University of Defense Technology, Changsha, Hu Nan, 410073, P. R. China
† Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, FL 33431

***Abstract*— The topological properties of peer-to-peer overlay networks are critical factors that dominate the performance of these systems. Several non-constant and constant degree interconnection networks have been used as topologies of many peer-to-peer networks. One of these has many desirable properties: the Kautz digraph. Unlike interconnection networks, peer-to-peer networks need a topology with an arbitrary size and degree, but the complete Kautz digraph does not possess these properties. In this paper, we propose MOORE: the first effective and practical peer-to-peer network based on the incomplete Kautz digraph with $O(\log_d N)$ diameter and constant degree under a dynamic environment. The diameter and average routing path length are $\lceil \log_d(N) - \log_d(1 + 1/d) \rceil$ and $\log_d N$, respectively, and are shorter than that of CAN, butterfly, and cube-connected-cycle, and are close to that of complete de Bruijn and Kautz digraphs. The message cost of node joining and departing operations are at most $2.5d \log_d N$ and $(2.5d + 1) \log_d N$, and only d and $2d$ nodes need to update their routing tables. MOORE can achieve optimal diameter, high performance, good connectivity and low congestion evaluated by formal proofs and simulations.**

Keywords: Constant degree networks, Kautz digraphs, peer-to-peer networks

I. INTRODUCTION

Structured peer-to-peer networks, abbreviated as P2P, have recently emerged as a good candidate infrastructure for building novel large-scale and robust network applications [1], [2], [3], [4], [5], [6] in which participating peers share resources as equals. In the past several years, various structured P2P overlay networks have been proposed, and more are likely to come. In general, the topological properties of structured P2P overlay networks are critical factors that dominate the performance of these systems. Therefore, it is very important to design a suitable topology for particular applications.

Several non-constant and constant degree topologies of interconnection networks have been used as the ideal topology in P2P networks. The degree and diameter increase logarithmically with respect to the size of the network for non-constant degree topologies, such as hypercube and ring digraph. The diameter increases logarithmically with respect to the size of the network, but the in-degree or out-degree of each vertex is a constant for constant degree topologies, such as cube-connected-cycle [7] (CCC), butterfly, d-dimensional torus, de Bruijn [8], and Kautz [9] digraph. Among existing P2P networks, Pastry [3] and Kademlia [4] are based on the hypercube topology, Viceroy [5] and Ulysses [10] are based on the butterfly topology, CAN [1] is based on the d-dimensional torus topology, Koorde [6], Distance Halving [11], D2B [12], [13], ODRI [14] and Broose [15] are based on the de Bruijn topology, and FissionE [16] is based on the Kautz topology.

It is well known that there are two important requirements for P2P network topologies. First, P2P networks always pursue a topology with arbitrary size and degree in order to deal with the uncontrolled dynamic operations of nodes, such as joining, departing and failing. Second, P2P networks attempt to design and implement a topology with the smallest diameter (the largest number of hops needed for the shortest routing path between a pair of source-destination nodes) possible given N nodes and fixed degree d (the size of routing table and links to be maintained on each node). Constant degree topologies can satisfy the second requirement, and the Kautz digraph can obtain a smaller diameter than other constant degree topologies with the same degree and order. Unfortunately, the orders of the Kautz digraph and many other constant topologies mentioned above are a series of discrete integers but cannot cover all integers under a given degree d. Therefore, they cannot satisfy the first requirement.

In this paper, we design an incomplete Kautz digraph with arbitrary network size and degree which can satisfy the above two requirements and still retain the key properties of a complete Kautz digraph. Then, we propose MOORE (this name implies that the network topology can almost achieve the *Moore bound* discussed in Sections II and VII): the first effective and practical peer-to-peer network based on the incomplete Kautz digraph with $O(\log_d N)$ diameter and constant degree under a dynamic environment. The diameter and average routing path length are $\lceil \log_d(N) - \log_d(1+1/d) \rceil$ and $\log_d N$, respectively, and are shorter than that of CAN, butterfly, and CCC but close to that of complete de Bruijn and Kautz digraphs. The message cost of node joining and departing operations are at most $2.5d \log_d N$ and $(2.5d+1) \log_d N$, respectively, and only d and $2d$ nodes need to update their routing tables. MOORE can achieve optimal diameter, high performance, good connectivity and low congestion.

The work of Deke Guo was supported in part by the National High Technology Research and Development Program of China under grants No. 2002AA131010 and 2003AA135110. E-mail: aeronautic@126.com. The work of Jie Wu was supported in part by NSF grants, ANI 0073736, CCR 0329741, CNS 0422762, CNS 0434533, and EIA 0130806. E-mail: jie@cse.fau.edu

The main contributions of this paper are the following:

1) We present the definition, construction procedure and theory results of an incomplete Kautz digraph with arbitrary order and degree which can satisfy the two important requirements and retain desirable properties of a complete Kautz digraph, such as optimal diameter, constant out-degree, simple routing scheme and low congestion.
2) We design a new structured peer-to-peer network, called MOORE, based on the incomplete Kautz digraph, and provide a suitable resource distribution policy, production methods of resource and node identifier, and a shortest path routing scheme.
3) We propose some relevant algorithms necessary to handle the uncontrolled dynamic operations of nodes, such as node joins and departs, and network expands and shrinks. These algorithms can preserve the desirable structure of backbone subnetwork and guarantee the correctness and performance of MOORE.
4) We evaluate the performance and cost of MOORE through formal analysis and simulation, and compare it with mainstream structured peer-to-peer networks based on other constant degree topologies.

The rest of this paper is organized as follows. Section II surveys the definition and emulation methods of the Kautz digraph. Section III proposes the theory of an incomplete Kautz digraph and its construction procedure. Section IV describes the detailed design of MOORE. Section V proposes the construction and maintenance algorithms of the topology. Section VI analyzes and evaluates the characteristics of MOORE. Our conclusions and future work are discussed in Section VII.

II. Related work

A. Kautz digraph

It is well known that a Kautz digraph can be defined in two different but equivalent ways: as digraphs on alphabets (the standard method) and using congruent arithmetic [17], [18].

Definition using alphabet: Let $Z_d = \{0, 1, ..., d\}$ be an alphabet of $d + 1$ letters, and $Z_d^D = \{x_1..x_{D-1}x_D \,|\, x_i \in Z_d, x_i \neq x_{i+1} \text{ and } 1 \leq i < D\}$ is a Kautz identifier space consisting of all Kautz identifiers with length D and base d. The vertex set and arc set of the Kautz digraph are Z_d^D and $E(K(d, D)) = \{\langle x_1x_2...x_D, x_2, ...x_D\alpha\rangle \,|\, \alpha \in Z_d, \alpha \neq x_D\}$.

Definition using congruent arithmetic: Let $GK(d, n)$ denote the generalized Kautz digraph with degree d and order n, respectively. The vertex set and arc set of the generalized Kautz digraph are $V(GK(d, n)) = \{0, .., n - 1\}$ and $E(GK(d, n)) = \{\langle i, (-d \times i - \alpha) \bmod n\rangle \,|\, 1 \leq \alpha \leq d\}$ [19], [20].

The order N of a digraph with maximum out-degree d and diameter D is bounded by the so-called *Moore bound* [21]:

$$N \leq d^D + d^{D-1} + ... + d^2 + d + 1 = (d^{D+1} - 1)/(d - 1). \quad (1)$$

The Moore bound is provably not achievable for any non-trivial digraph. Kautz digraphs come close to the Moore bound and can be built with $N = d^D + d^{D-1}$ nodes. P2P networks are always concerned with the *order/diameter problem*: Given N nodes and a fixed degree d, what is the minimum diameter? The following lower bound can be derived from (1):

$$D \geq \lceil \log_d (N(d - 1) - 1)\rceil - 1. \quad (2)$$

B. Emulation of Kautz digraph

The topology is incrementally extendable if its definition allows graphs of arbitrary size and degree. According to the above definition, the Kautz digraph is not incrementally extendable. The generalized Kautz digraph can be defined for any number of vertices, but it is also not incrementally extendable because its index of expandability[1] is too large, proportional to the number of arcs [17], [18].

The most related research work revolves around FISSIONE, which uses a Kautz graph $K(2, k)$ as its static topology and proposes some emulation methods of $K(2, k)$ to deal with the dynamic operations of nodes. The topology of FISSIONE does not support graphs of arbitrary degree, and is not a definite constant degree digraph because it is regular , and not out-degree regular. Furthermore, the emulation methods of $K(2, k)$ are not suitable to a general Kautz graph $K(d, k)$. Thus, FISSIONE is not incrementally extendable.

III. Incomplete Kautz Digraph

A. Incomplete Kautz digraph

Let $G = (V, E)$ be a strongly connected digraph. The vertex set and arc set are denoted as $V = V(G)$ and $E = E(G)$, respectively. An arc from vertex u to v is denoted $\langle u, v\rangle$. The arc is said to be incident from vertex u and incident on vertex v. The set of vertices incident on vertex u is denoted as $\Gamma_G^-(u) = \{v \in V(G) \,|\, \langle v, u\rangle \in E(G)\}$, and $\delta_G^-(u) = |\Gamma_G^-(u)|$ is the in-degree of vertex u. Similarly, the set of vertices incident from u is denoted as $\Gamma_G^+(u) = \{v \in V(G) \,|\, \langle u, v\rangle \in E(G)\}$, and $\delta_G^+(u) = |\Gamma_G^+(u)|$ is the out-degree of vertex u.

Definition 1: Let digraph G be a complete Kautz digraph $K(d, D)$, and $E' \subseteq E(G)$ be a subset of arcs which are incident from all vertices of G, that is, $\{u : \langle u, v\rangle \in E'\} = V(G)$. A digraph of fixed out-degree d and order n, $IK(d, n)$, is an incomplete Kautz digraph only if the following conditions hold, and G is the predecessor Kautz digraph of $IK(d, n)$.

1) Vertices of $IK(d, n)$ represent the arcs of E', that is, $V(IK(d, n)) = \{uv; \langle u, v\rangle \in E'\}$, and $|E'| = n$;
2) Vertex uv of $IK(d, n)$ is adjacent to the vertices $v'w$, for each $w \in \Gamma_G^+(v)$, where

$$v' = \begin{cases} v, & vw \in E' \\ \text{any vertex of } \Gamma_G^-(w), & \text{otherwise.} \end{cases} \quad (3)$$

According to Definition 1, any arc $\langle u, v\rangle$ of $K(d, D)$ can be denoted as a vertex labeled $uv = u_1u_2u_Dv_D$ of $IK(d, n)$. In this paper, we will not distinguish strictly between an arc of $K(d, D)$ and its corresponding vertex in $IK(d, n)$. For example, we may use $\langle u, v\rangle$ to denote a vertex of $IK(d, n)$. The

[1] The index of expandability is the minimum number of arcs that have to be deleted from $IK(d, n + 1)$ to obtain a subgraph of $IK(d, n)$.

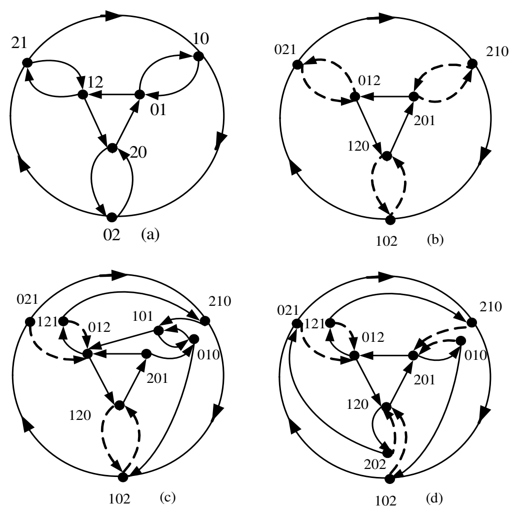

Fig. 1. A complete Kautz digraph $K(2,2)$, and incomplete Kautz digraphs $IK(2,6), IK(2,9)$ induced by the factorization of $K(2,2)$.

arc $\langle uv, vw\rangle$ will be called an α-arc if $vw \in E'$. Otherwise, we say that $\langle uv, v'w\rangle$ is a β-arc. It is straightforward that the out-degree of any vertex of $IK(d,n)$ is d.

According to Definition 1, it is straightforward to design an incomplete Kautz digraph $IK(d,n)$ through the following general construction procedure:

1) Discover the largest complete Kautz digraph $K(d,D)$ satisfying $d^D + d^{D+1} < n$.
2) Construct a subset E' of $E(K(d,D))$ and $E' = n$, such that each vertex of $K(d,D)$ is covered by at least one arc of E'.
3) Produce all vertices of $IK(d,n)$ by presenting each arc of E' as a vertex. Then, establish links among vertices according to the constraint (3) mentioned above.

The general construction procedure can produce many different incomplete Kautz digraphs with same order for different arc set E', but only ensures that the minimum *in-degree* of the resulting incomplete digraphs is not less than 1. Thus, this procedure alone is not strong enough to produce an incomplete digraph that inherits the desirable properties of the complete one in a deterministic manner. Therefore, a method for careful selection of arc set E' is necessary.

B. Construction of incomplete Kautz digraph

Let $G = (V,E)$ be a strongly connected digraph. An arc a covers a vertex x if a is incident from x. An arc set $E' \subset E$ is an *arc-covering* of G if every vertex of G is covered by at least one arc of E'. If $|E'| = |V|$, E' is called a *1-arc-covering*. If $\forall u \in V$; $\delta^-_{G'}(u) = \delta^+_{G'}(u) = 1$ for $G' = (V,E')$, then E' is called a *1-factor* of G. Hence, a 1-factor is a spanning 1-regular subdigraph and consists of cycles and possibly loops. A digraph G has a 1-factorization if its arc set can be partitioned into some arc-disjoint 1-factors.

Definition 2: Let $Lshift$ denote a binary operation such that $Lshift(x_1...x_{D-1}x_D, i) = x_1...x_{D-1}x'_D$. If $(x_{D-1} + i - d - 1) < x_{D-1} < x_D$ or $x_{D-1} > x_D$ and $x_{D-1} > x_D + i$, then $x'_D = (x_D + i) \bmod (d+1)$. Otherwise, $x'_D = (x_D + i + 1) \bmod (d+1)$ [18].

Definition 3: Let $Rshift$ denote a binary operation such that $Rshift(x_1x_2...x_{D-1}x_D, i) = x'_1x_2...x_{D-1}x_D$. If $x_2 + i - d - 1 < x_1 < x_2$ or $x_1 > x_2$ and $x_1 - i > x_2$, then $x'_1 = (x_1 - i) \bmod (d+1)$. Otherwise, $x'_1 = (x_1 - i - 1) \bmod (d+1)$.

Definition 4: For $x = x_1x_2..x_D \in V(K(d,D))$ and $0 \leq i \leq d-1$, the left k-shift operation and right k-shift operation, denoted as σ^i_k and σ^{-i}_k, respectively, are defined as follows:

$$\sigma^i_1(x) = \begin{cases} Lshift(x_2..x_Dx_1, i), \text{ if } x_1 \neq x_D \\ Lshift(x_2..x_Dx_2, i), \text{ if } x_1 = x_D \end{cases} \quad (4)$$

$$\sigma^i_k = \sigma^i_1 \circ \sigma^i_{k-1} \quad (5)$$

$$\sigma^{-i}_1(x) = \begin{cases} Rshift(x_Dx_1..x_{D-1}, i), \text{ if } x_1 \neq x_D \\ Rshift(x_{D-1}x_1..x_{D-1}, i), \text{ if } x_1 = x_D \end{cases} \quad (6)$$

$$\sigma^{-i}_k = \sigma^{-i}_1 \circ \sigma^{-i}_{k-1}. \quad (7)$$

For vertex x, vertex $\sigma^i_1(x)$ and vertex $\sigma^{-i}_1(x)$ are its $(i+1)^{th}$ successor and predecessor, respectively. Furthermore, $\langle x, \sigma^i_1(x)\rangle$ and $\langle \sigma^{-i}_1(x), x\rangle$ denote its $(i+1)^{th}$ *out-arc* and *in-arc*. In fact, the $(i+1)^{th}$ *out-arc* and *in-arc* of each vertex are unique under the σ^i_1 operation and σ^{-i}_1 operation.

Theorem 1: The arc set $E(K(d,D))$ can be partitioned into d arc-disjoint 1-factors $\digamma_0, ..., \digamma_{d-1}$ under corresponding operation σ^i_1, such that $K(d,D)$ has a 1-factorization.

Proof: Let any vertex, as the beginning point, take a walk through $K(d,D)$. For each vertex x under this walk, it always walks along the $(i+1)^{th}$ *out-arc* $\langle x, \sigma^i_1(x)\rangle$ under left shift operation σ^i_1. The walk will meet a covered vertex after at most $d^D + d^{D-1}$ steps. This walk will not meet any inner vertex because the $(i+1)^{th}$ *in-arc* of each inner vertex in the walk is unique and has been used by its predecessor in this walk. Therefore, this walk will get back to the beginning vertex along its $(i+1)^{th}$ *in-arc*, and finally form a cycle.

According to the above discussions, each vertex of $K(d,D)$ is covered by at least one cycle under the operation σ^i_1. Let us suppose there is a common vertex y covered by a pair of cycles under operation σ^i_1. It is easy to conclude that the two cycles must also cover the vertex satisfying the fact that its $(i+1)^{th}$ out-arc is incident on vertex y. From point of recursive operation, we can conclude that the two cycles are identical. Therefore, each vertex is covered by only one cycle under operation σ^i_1, and cycles are mutually vertex disjointed. The cycles under operation σ^i_1 form a spanning 1-regular subdigraph, and produce a 1-factor $\digamma_i$ of $K(d,D)$. Furthermore, for any vertex x of $K(d,D)$ the arc covering it is different for a different 1-factor. Therefore, those 1-factors are mutually arc-disjoint, and $K(d,D)$ has a factorization. Therefore, Theorem 1 holds. ■

Corollary 1: The identifier of 1-factor containing the corresponding arc of any vertex x of $IK(d,n)$ is determinate.

Proof: According to Algorithm 1, the corresponding arc in $K(d,D)$ of vertex $x = x_1...x_Dx_{D+1}$ belongs to the 1-factor labeled $\digamma(x) =$ Distance $(\sigma^0_1(x_1x_2..x_D), x_2x_3..x_{D+1})$. ■

Theorem 2: The incomplete Kautz digraph $IK(d,n)$ induced by any k 1-factors of $Kautz(d,D)$ is a d-regular digraph for all $1 \leq k \leq d$, where $n = k \times (d^D + d^{D-1})$.

Algorithm 1 Distance(y,z)

Require: y and z are different d-ary Kautz identifiers with length $D+1$.

if $D = 0$ **then**
  $j \leftarrow (z_{D+1} - y_{D+1}) \bmod (d+1) - 1$
else
  if $\min(y_{D+1}, z_{D+1}) < y_D < \max(y_{D+1}, z_{D+1})$ **then**
    if $z_{D+1} > y_{D+1}$ **then**
      $j \leftarrow z_{D+1} - y_{D+1} - 1$
    else
      $j \leftarrow z_{D+1} - y_{D+1} + d + 1$
  else
    if $z_{D+1} > y_{D+1}$ **then**
      $j \leftarrow z_{D+1} - y_{D+1}$
    else
      $j \leftarrow z_{D+1} - y_{D+1} + d$
return j

Proof: Each vertex x of $K(d, D)$ is covered by an arc $\langle x, \sigma_1^i(x)\rangle$ of 1-factor F^i. According to Definition 1, the vertex labeled $\langle x, \sigma_1^i(x)\rangle$ is incident on other d vertices of the $IK(d, d^D + d^{D+1})$ induced by F^i. This proves that the incomplete Kautz digraph induced by F^i is d-out-regular.

On the other hand, vertex $\langle x, \sigma_1^i(x)\rangle$ is incident from vertex $\langle \sigma_1^{-i}, x\rangle$ through an α-arc in the incomplete Kautz digraph induced by F^i, because the arc $\langle x, \sigma_1^i(x)\rangle$ is incident on the arc $\langle \sigma_1^{-i}, x\rangle$ in a cycle of F^i. Furthermore, vertices $\sigma_1^{-j}(\sigma_1^i(x))$ for $0 \le j \le d-1$ and $j \ne i$ are incident on the vertex $\sigma_1^i(x)$ and covered by arcs $\langle \sigma_1^{-i}(\sigma_1^{-j}(\sigma_1^i(x))), \sigma_1^{-j}(\sigma_1^i(x))\rangle$ of F^i. This proves that the incomplete Kautz digraph induced by F^i is a d-in-regular and d-regular digraph.

The union of any k 1-factors, $1 \le k \le d$, also produces a d-regular incomplete Kautz digraph $IK(d, k \times (d^D + d^{D-1}))$ according to similar reasoning, but the number of α-arcs and β-arcs are k and $(d-k)$, respectively, among the d out-arcs and d in-arcs of each vertex. Therefore, Theorem 2 holds. ∎

The general construction method of $IK(d, n)$ does not propose any method for selection of arc set E'. Random selection cannot ensure that the connectivity of an incomplete Kautz digraph is close to that of its complete Kautz digraph predecessor. We will use results of Theorem 1 and Theorem 2 to construct the arc set E', and enable the resulting $IK(d, n)$ to achieve better connectivity. The ideal arc set E' and $IK(d, n)$ can be achieved by following a special construction procedure based on the 1-factorization of $K(d, D)$:

1) In order to construct a $IK(d, n)$ where $k(d^D + d^{D-1}) \le n \le (k+1)(d^D + d^{D-1})$, we start with a d-regular incomplete Kautz digraph $IK(d, d^D + d^{D-1})$ induced by one 1-factor F^i of $K(d, D)$ through Algorithm 3. $K(d, D)$ can be achieved from a initial small complete Kautz digraph by invoking this procedure repeatedly.
2) Then, add vertices corresponding to arcs of other $k-1$ 1-factors to the d-regular digraph mentioned above, and achieve a new d-regular digraph $IK(d, k(d^D + d^{D-1}))$. This can be realized by using Algorithm 4 recursively.
3) Then, add vertices corresponding to $n - k(d^D + d^{D-1})$ arcs, denoted $F^{k'}$, of another 1-factor F^k to the new d-regular digraph. This process can also be realized by using Algorithm 4 recursively.

The last step in the above procedure is based on proper choice of the added arcs as discussed in Section IV. In order to achieve higher connectivity, the arc selection polices must make the minimum in-degree of the final digraph as larger as possible. Theorem 3 shows the bounds of minimum in-degree of a $IK(d, n)$ that resulted from above procedure.

Theorem 3: For all k, $1 \le k < d$, and for all n such that $k(d^D + d^{D-1}) \le n \le (k+1)(d^D + d^{D-1})$, any incomplete Kautz digraph $IK(d, n)$ satisfied that $k \le \delta^-(IK(d, n)) \le d$.

Proof: We know that the number of 1-factors of $K(d, D)$ used to produce the $IK(d, n)$ is $k+1$. For the sake of generality, we select the first $k+1$ 1-factors $F^0, F^1 ..., F^k$, but the result is same for any $k+1$ 1-factors. The special construction procedure can produce the needed incomplete Kautz digraph mentioned in this theorem. Theorem 2 can also guarantee that the incomplete Kautz digraph induced by any k 1-factors of $K(d, D)$ is a d-regular digraph.

The adding operation of any vertex x induced by $F^{k'}$ mentioned above has an effect on one out-arc of at most d existing nodes. Node x needs to inform its predecessor $\sigma_1^{-i}(x)$ for $0 \le i \le k-1$ to update its $(i+1)^{th}$-out-arc (a β arc) with a new α-out-arc incident on node x. This also results in the in-degree of the node at other end of the original $(i+1)^{th}$-out-arc of node $\sigma_1^{-i}(x)$ decreasing by one. If the arc corresponding to its predecessor $\sigma_1^{-k}(x)$ has been added previously, node x also informs this predecessor to add an α-arc to itself. For $k+1 \le i \le d-1$, other $d-k-1$ predecessors of node x are induced by 1-factors F^i and do not exist in $IK(d, n)$, but there may exist other l nodes corresponding to arcs $\langle \sigma_1^{-k}(\sigma_1^{-i}(x_2..x_{D+1})), \sigma_1^{-i}(x_2..x_{D+1})\rangle$ of $F^{k'}$, which are incident on node x through a β arc and $0 \le l \le d-k-1$.

According to the above analysis, the in-degree of vertices induced by $F^{k'}$ should be at least k and less than d, unless $k = d-1$ and arcs of F^i forms cycles. The in-degree of vertices induced by previous k 1-factors should not be less than $d-1$, and can reach d at some scenarios such as Figure 1(d). Thus, $k \le \delta^-(IK(d, n)) \le d$, and Theorem 3 holds. ∎

IV. MOORE DESIGN

A. Overview

To organize peers in an efficient overlay network, a structuring strategy that is easy to understand and implement is required. Typically, a structured P2P overlay network is built such as to guarantee logarithmic diameter while maintaining a compact routing table of logarithmic or constant size. An incomplete Kautz digraph inherits many desirable characteristics of a complete one, and is more practical than a complete one because its order can be of an arbitrary size. Therefore, MOORE selects an incomplete Kautz digraph over a complete one as its topology in a dynamic environment.

In this paper, we use two Kautz identifier spaces $Z_d^l = \{x_1..x_{l-1}x_l \,|x_i \in \{0, 1, .., d-1\}\}$ and Z_d^m as the *resources* identifier space and *nodes* identifier space of MOORE. The length of the resource identifier should be larger than that of the node identifier, but not necessarily too much larger. If we fix the out-degree d of MOORE, then we can infer that $m = \lceil \log_d^{n_n} - \log_d^{(1+1/d)} \rceil$ and $l = \lceil \log_d^{n_r} - \log_d^{(1+1/d)} \rceil$ where n_n and n_r denote the maximum number of nodes and resources, respectively, of MOORE.

Furthermore, we also need to consider the policy for distributing resources among nodes of MOORE. In the case of a complete Kautz digraph, the resource with identifier $x_1x_2..x_l$ is stored and maintained by node labeled $y_1y_2..y_m$ if and only if $y_1y_2..y_m$ is a prefix of $x_1x_2..x_l$. This is the same as in the case of an incomplete Kautz digraph if the node labeled $y_1y_2..y_m$ exists in the digraph. Otherwise, the resource will be taken over by another node corresponding to an arc $\langle y_1y_2..y_{m-1}, \sigma_1^k(y_1y_2..y_{m-1})\rangle$ in $k(d, m-1)$. According to Definition 4 and Theorem 1, k denotes the identifier of the 1-factor that was selected to induce the incomplete Kautz digraph with the same order as $K(d, m-1)$, and the default value of k is 0 in general.

B. Mapping resource onto resources identifier space

Each resource accessible through MOORE will receive an identifier taken from Z_d^l, and different resources are allowed to receive the same identifier. The mapping of resources onto Z_d^l can be implemented in several ways. Literature [16] proposed a determinate algorithm to generate an identifier with base 2 for each resource. In reality, the base of an incomplete Kautz digraph used by MOORE is often larger than 2 for the sake of decreasing its diameter and improving its connectivity. Therefore, this paper considers another determinate $Kautz\_hash$ algorithm to generate an identifier with any base for each resource. The $Kautz\_hash$ uses three parameters: key denotes the original identifier of resource such as name or keyword; d and l denote the base and length of expected Kautz strings, respectively. $Kautz\_hash$ is detailed below.

First of all, it achieves a binary string with a larger length by hashing the key according to a given consistent hash table such as $SHA-1$. Then, it converts the resulting binary string to a new string S_0 with base d, and substitutes all substrings consisting of any identical number with a single one. If the length of S_0 is less than l, it appends $i = 1$ to key and achieves a new Kautz string S_i with base d, and then appends S_i to S_0. If the length of S_0 is still less than l, it appends the value of $i+1$ to key and repeats the procedure again until the length of S_0 becomes larger than l. Finally, the substring consisting of the first l numbers of S_0 from left to right is returned as the identifier of the resource.

C. Mapping node onto nodes identifier space

In practice, MOORE starts with $d^{m_0} + d^{m_0-1}$ initial nodes and forms a structured P2P network according to a complete Kautz digraph $K(d, m_0)$, then enlarges or shortens its scale through a series of dynamic operations at run time. Thus, the nodes' identifier space should not be a static one compared to the resources' identifier space. It should start with an initial identifier space, then is enlarged or shortened with the increase or decrease of MOORE scale, respectively. The initial identifier space is $Z_d^{m_0}$ where $m_0 < m$, and each identifier of this space will be allocated to a unique node. If all identifiers of $Z_d^{m_0}$ were allocated and new nodes apply to participate in the initial system, the initial nodes' identifier space should be extended to $Z_d^{m_0+1}$ and allocate free identifiers to new nodes. Note that the new identifier space is a d multiple of the old one and can be achieved according to Definition 1.

As a direct result of this operation, the original identifiers of initial nodes also need to be updated by the first $d^{m_0} + d^{m_0-1}$ new identifiers induced by the 1-factor F^0 of $K(d, m_0)$, then the initial nodes form another d-regular incomplete Kautz digraph $IK(d, d^{m_0} + d^{m_0-1})$ according to Algorithm 3. In order to maintain better structuring properties under a dynamic environment, we must focus on the policy used to allocate identifiers to new nodes, and this policy is equivalent to the arc choice policy used by the special construction procedure of the incomplete Kautz digraph mentioned above. Any arc choice policy takes first arcs of the second 1-factor F^1, then arcs of the third 1-factor F^2, and so on. But existing policies are different in the selection order of arcs in each 1-factor.

The arc choice policy proposed in literature [17], [18] suggests to take first arcs of one cycle in each 1-factor, then arcs of the second cycle, and so on. The random choice policy, denoted as $factorRandom$, selects arcs randomly from given 1-factor. The difference between these two policies is that the former can make the in-degree of more new vertices reach $k+1$ than the latter. In this paper, we propose an enhanced policy denoted as *cycleSequence* , which takes arcs of one cycle along its direction continuously, then the second cycle, and so on. Our new policy can make more vertices reach $k+1$ in-degree than the policy proposed in literature [17], [18], because the $(k+1)^{th}$ predecessor of a newly added arc has been added previously except if it is the first selected arc of a cycle. The k satisfies that $(d^{m_0} + d^{m_0-1}) \leq n \leq (k+1) \times (d^{m_0} + d^{m_0-1})$, and n denotes the number of existing nodes in MOORE.

Let n denote the number of nodes or allocated identifiers. Recall that each new added node x can result in the in-degree of at most k nodes induced by previous k 1-factors and incident from the old $(k+1)^{th}$-out-arc of its predecessor $\sigma_1^{-i}(x)$ decreases by one, where $0 \leq i \leq k-1$ and k is the largest number such that $d^k + d^{k-1} \leq n$. As an example, if we add new vertex 121 to the $IK(2, 6)$ induced by 1-factor F^1 of $K(2, 3)$ in Figure 1(c), the original β-out-arc from vertex 012 to 021 will be updated with a α-out-arc from vertex 012 to 121. Thus the in-degree of vertex 021 decreases by one. No existing arc choice policies focus on this problem. Therefore, we propose a different policy denoted as *inDegreePreserved* to deal with it. The basic idea is to allocate the identifier of the $(k+1)^{th}$ predecessor of existing nodes once their $(k+1)^{th}$-in-arc is canceled by previous node's adding operation, and reestablish its $(k+1)^{th}$-in-arc with a α-arc incident from its $(k+1)^{th}$ predecessor. This policy tries to preserve the in-degree-regularity of nodes induced by previous k 1-factors, and is very efficient if $k = d-1$ or $d = 2$. Thus, MOORE can achieve the best structuring properties if it combines the policies *inDegreePreserved* and *cycleSequence*.

On the other hand, an identifier allocated to a node may become a free identifier if the node failed or departed from the network and did not recover during a given time interval. All arc choice policies should give these kinds of identifiers priority when they allocate an identifier to a new node. If this identifier is induced by previous F^i for $0 \leq i \leq k-1$, this operation is helpful to preserve the desirable structure of the backbone subnetwork consisting of nodes induced by previous k 1-factors. Otherwise, this operation can make the in-degree of more nodes reach $k+1$ for the *cycleSequence* policy.

D. Routing scheme

In order to route messages to destinations correctly, each node x of MOORE must establish links with selected neighbors and construct a routing table according to Definition 1 and Algorithm 4, and update its links and routing table when other nodes join, depart or fail. The routing table consists of d entries, and each entry includes the identifier and address (such as IP and port number) of one neighbor node. Furthermore, node x may initiate a *lookup* message to find a given resource or node with identifier y, and initiate a *insert* message to distribute its resource with identifier y to a responsible node. We propose Algorithm 2 to route these kinds of messages to the destination node along the shortest path.

Algorithm 2 Route(y, message, scheme)

Require: Identifier y is not less than x
$z \leftarrow y$
if the length of y is larger than D **then**
  $y \leftarrow y_1 y_2 .. y_D$
if $x = y$ or $x_1 x_2 .. x_{D-1} = y_1 y_2 .. y_{D-1}$ **then**
  Process the message locally, and return $success$.
$x' \leftarrow$ forward_orientation(y)
if $x' \neq null$ **then**
  return x'.Route(z, message, scheme)
else
  return $failure$ to the source node.

forward_orientation(y)
Let u be the largest integer such that $x_{D-u+i} = y_i$ for $1 \leq i \leq u$, and $result \leftarrow null$
for $i = 0$ to d **do**
  $w \leftarrow routingtalbe[i].identifier$
  if $u = 0$ and $w = y$ **then**
    return w
  else if $w_{D-u-1+i} = y_i$ for $1 \leq i \leq u+1$ **then**
    $result \leftarrow w$
if $result = null$ and $scheme = resource$ **then**
  return $\sigma_1^k(x)$
else
  return $result$

Fiol proposed a method to achieve a short path from x to y in [22]: find the largest suffix u of x that coincides with a prefix of y, then walk towards a neighbor z of x such that its largest suffix v coincides with a prefix of y and the length of v is larger than that of u. Note that the exhibited path does not necessarily have the shortest length, because of the β-out-arc. As an example, consider the graph in Figure 1(c). Suppose node 021 needs to route to node 012 along the short path $021 \rightarrow 210 \rightarrow 101 \rightarrow 012$, but the shortest path should be $021 - 012$ resulting from a β-out-arc incident from node 021. In order to deal with this problem, Algorithm 2 will check whether there is a routing entry corresponding to node y if the length of u is zero. Note that, Algorithm 2 can also achieve similar lower congestion as the long path routing scheme [9], [16], this will be proved by our simulation results.

Algorithm 2 uses three parameters: y denotes the identifier of a target resource or node; $message$ denotes the real message needed to be routed; $scheme$ denotes the type of message, and can be $resource$ (lookup or insert resource) or $node$ (find the address of node). Recall that the resource distribution policy of an incomplete Kautz digraph is different from that of the complete one, because any resource has two possible exclusive destination nodes. Therefore, if $scheme = resource$ and the method forward_orientation in Algorithm 2 does not find the node whose identifier is a prefix of the identifier of target resource, it will forward the message to another destination node defined by the resource distribution policy mentioned above.

V. Topology construction and maintenance of MOORE

MOORE selects incomplete Kautz digraph as its topology, and its topology can evolve from an initial Kautz digraph in a distributed manner by using Definition 1 recursively. The initial Kautz digraph can be constructed through many mature centralized methods, so we do not focus on it in this paper. In practice, MOORE needs to deal with the following operations: node joins, node departs, network expands, and network shrinks. It is these operations that drive the evolution of the MOORE topology. This section proposes some relevant algorithms necessary to implement these operations.

A. Topology expands

We know that the topology of MOORE is a $IK(d,n)$, and n is covered by an unique range $[d^D + d^{D-1}, d^{D+1} + d^D)$. In practice, the topology will become a complete Kautz digraph if n reaches the upper boundary of this range. In this situation, if other nodes apply to join MOORE, it needs to expand the topology to a new incomplete Kautz digraph with order of n equal to the lower boundary of a new range $[d^{D+1} + d^D, d^{D+2} + d^{D+1})$. The expanding operation includes at least the following two steps. First, each existing node needs to update its original identifier according to Definition 1 with the 1-factor F^0 of $K(d,D)$ as the arc set E'. Second, all existing nodes form a new structured P2P network according to the new topology $IK(d,n)$. These operations can be implemented by following Algorithm 3. The parameter k used by Algorithm 3 denotes identifier of the 1-factor that was selected to induce the incomplete Kautz digraph with the same order as $K(d,D)$, and the default value of k is 0.

B. Node joins

As for most P2P networks, we assume there are some existing nodes as *entry points* of MOORE, which can receive

Algorithm 3 Extend $(K(d, D), k)$

Require: $K(d, D)$ is a d-regular complete Kautz digraph with diameter D. And $0 \leq k < d$.

for each node x labeled $x_1x_2..x_D$ in $K(d, D)$ **do**
  $x.label \leftarrow \langle x, \sigma_1^k(x) \rangle$
  Node x constructs a temporary routing table.
  $y = y_1y_2..y_{D+1} \leftarrow \langle \sigma_1^k(x), \sigma_2^k(x) \rangle$
  for $i = 0$ to $d - 1$ **do**
    if $k = i$ **then**
      $z = z_1z_2..z_{D+1} \leftarrow \langle \sigma_1^k(x), \sigma_2^k(x) \rangle$
      $address \leftarrow x.routing[k].address$
    else
      $z = z_1z_2..z_{D+1} \leftarrow \langle \sigma_1^{-k}(\sigma_1^i(\sigma_1^k(x)), \sigma_1^i(\sigma_1^k(x)) \rangle$
      $address \leftarrow$ Route($\sigma_1^{-k}(\sigma_1^i(\sigma_1^k(x)),$, node))
    $j \leftarrow$ Distance(y, z)
    Node x adds $\langle z, address \rangle$ as the $(j + 1)^{th}$ entry of the temporary routing table.
for each node x in $K(d, D)$ **do**
  Updates its routing table with the temporary routing table, then updates links according to new routing table.

Algorithm 4 Node joins(x,y,$label$)

$k \leftarrow \mathcal{F}(x)$
for $i = 0$ to d **do**
  if $i \leq label$ and node $\langle x_2x_3..x_{D+1}, \sigma_1^i(x_2x_3..x_{D+1}) \rangle$ exists **then**
    Node y finds the $address$ of node z. Then node x adds $\langle z, address, \alpha \rangle$ as its $(i + 1)^{th}$ routing entry, and establishes a link to node z, where $z = \langle x_2x_3..x_{D+1}, \sigma_1^i(x_2x_3..x_{D+1}) \rangle$,
  else
    Node x asks node y to find the $address$ of node z labeled $\langle \sigma_1^{-k}(\sigma_1^i(x_2x_3..x_{D+1})), \sigma_1^i(x_2x_3..x_{D+1}) \rangle$
    if node z does not exist **then**
      Node x asks node y to find the address of a node z labeled $\langle \sigma_1^{-j}(\sigma_1^i(x_2x_3..x_{D+1})), \sigma_1^i(x_2x_3..x_{D+1}) \rangle$, where j is a random integer such that $0 \leq i \leq label$ and can node z exists.
    Node x adds $\langle z, address, \beta \rangle$ as the $(i + 1)^{th}$ entry of its routing table, and establishes a link to node z.
for $i = 0$ to d **do**
  if $i \leq label$ **then**
    $w \leftarrow \langle \sigma_1^{-i}(x_1x_2..x_D), x_1x_2..x_D \rangle$
  else
    $w \leftarrow \langle \sigma_1^{-k}(\sigma_1^{-i}(x_2..x_{D+1})), \sigma_1^{-i}(x_2..x_{D+1}) \rangle$
  Node w updates one original β link with an α or β link incident on node x, then updates its routing table.
Node x gets resources satisfied that x is their prefix of identifier from node $\langle x_1x_2..x_D, \sigma_1^0(x_1x_2..x_D) \rangle$.

and process the node joining message. The joining procedure includes three stages: receives a node identifier; redistribute resources; update routing tables. These operations can be implemented by following Algorithm 4.

Let $x_1x_2..x_{D+1}$ denote a node joining MOORE, and $y_1y_2..y_{D+1}$ denote an entry point of MOORE. Node x achieves its node identifier and identifier $label$ of current 1-factor according to the management policy of *nodes* identifier space. In reality, there exist at least two cases of node joining operation. The first case is $\mathcal{F}(x) = label$, which means that the new node belongs to the current 1-factor $\mathcal{F}^{label}$. The second case is $\mathcal{F}(x) < label$, which means that the new node belongs to the previous 1-factor and a node with the same identifier has either joined MOORE but failed or departed.

In both cases, node x needs to first find its successors and establish links and a routing table, then inform at most d existing predecessors to update their links and routing table, and finally take over its responsible resources from one existing node. The detailed process has been proposed when proving Algorithm 3. The $(i + 1)^{th}$ predecessor and successor of node x exist for $0 \leq i \leq k - 1$. Furthermore, its $(k + 1)^{th}$ successor does not exist unless it is the last arc of the current cycle, and its $(k + 1)^{th}$ predecessor exists unless it is the first arc selected from a cycle. Other j^{th} predecessors of node x do not exist for $k + 1 < j \leq d$, and it needs to find a substitute from nodes belonging to 1-factor $\mathcal{F}^{label}$, even nodes belonging to previous 1-factors in order to keep constant out-degree. So do other successors, but they do not find a substitute from nodes belonging to previous 1-factors in order to keep their connectivity.

C. Node departs

Let x denote a node departing from MOORE, and $label$ denote the identifier of current 1-factor. In practice, there exist at least two cases of node departing operation. The first case is $\mathcal{F}(x) = label$, which means that node x belongs to the current 1-factor $\mathcal{F}^{label}$. $\mathcal{F}(x) < label$ is another case, which means that node x belongs to the previous 1-factors. The node departing operation harms the topology structure and results in unsuccessful message routing. Algorithm 5 can compensate for the negative impact of the node leaving operation.

In the first case, node x needs to inform its in-neighbors to update the link incident on node x with another link incident on another node, and transfer its resources to another responsible node defined by the resource distribution policy mentioned above. In the second case, node x needs to find a substituted node y to replace it, and informs the in-neighbors of node y to update related links and routing entries. Then, node y takes over the identifier, resources, links and routing table of node x and its original identifier becomes free. Finally, node y updates its links according to the new routing table and informs its in-neighbor about its change of address. Note that node y should select first from nodes belonging to 1-factor $\mathcal{F}^{label}$, then nodes belonging to 1-factor $\mathcal{F}^{label-1}$ and so on. This policy can preserve the structure of backbone subnetwork consisting of nodes belonging to previous 1-factors before $\mathcal{F}^{label}$.

D. Topology shrinks

We also need to consider the topology shrinking operation when the number of nodes decreases to an order of the predecessor Kautz digraph. Let $x_1x_2...x_D$ denote any existing node, node x just needs to update its identifier as $x_2x_3...x_D$, and update the identifier of each routing entry in the same way. The implementation of this operation is simple and results in the least overhead. As an example, Figure 1(b) becomes Figure 1(a) through this operation.

Algorithm 5 Node departs $(x, label)$

if $F(x) < label$ **then**
  $y \leftarrow findSubstitute(x)$
  update$(y, label, F(x))$
  Node x transfers its resources and routing table to node y, then departs from MOORE. Node y updates its identifier, routing table, and links with that of node x, and informs in-neighbors about its change of address.
else
  Node x transfers its resources to node corresponding to arc $\langle y_1y_2..y_{m-1}, \sigma_1^0(y_1y_2..y_{m-1})\rangle$ before departing.
  update$(x, label, F(x))$

update$(z, label, l)$

for $i = 0$ to d **do**
  if $i < label$ **then**
    $w \leftarrow \langle\sigma_1^{-i}(z_1z_2..z_D), z_1z_2..z_D\rangle$
    Informs node w to update the link to node x with a new β link to node $\langle\sigma_1^i(z_2z_3..z_{D+1}), z_2z_3..z_{D+1}\rangle$.
  else
    $w \leftarrow \langle\sigma_1^{-l}(\sigma_1^{-i}(z_2..z_{D+1})), \sigma_1^{-i}(z_2..z_{D+1})\rangle$
    Informs node w to update the link to node x with a new β link to node $\langle\sigma_1^j(z_2z_3..z_{D+1}), z_2z_3..z_{D+1}\rangle$, where j is a random integer satisfied $0 \le j < label$ such that the new destination node exists.

VI. Analysis and Evaluation

We use PeerSim to implement MOORE. PeerSim is a P2P simulation framework aimed at developing and testing any kind of P2P protocols in a dynamic environment [23]. Our simulations are cycle based, and the MOORE topology with any size is evolved from the smallest Kautz digraph $K(d, 1)$ through those dynamic operations of nodes mentioned above. In this section, we will evaluate the following characteristics of MOORE: degree distribution, diameter, average path length, and congestion. The value of each characteristic under different network configurations is the average value of a sample achieved from at least 100 round simulations.

A. Degree distribution of MOORE

Corollary 2: MOORE is d-regular and has constant degree if its order equals to k multiple of n_0 for $1 < k \le d$ where n_0 denotes the order of its predecessor Kautz digraph. Otherwise, it is d-out-regular and has constant degree. Its index of expandability is not larger than $\delta^-(IK(d, n))$.

Proof: The proof has been proposed in Section III. ■

Theorem 3 proposes the bound on its minimum in-degree. In this section, we focus on the in-degree distribution of MOORE with order 7680 and 18000 under node identifier choice policies $factorRandom$ and $cycleSequence$.

Figure 2 shows the in-degree of most nodes are adjacent to d, and that of the remaining nodes are close to the trails of its in-degree distribution figure. The in-degree of more nodes are close to d and far away from the trails of its in-degree distribution if MOORE adopts the node identifier choice policy $cycleSequence$ rather than $factorRandom$ policy. Thus, $cycleSequence$ is more suitable to MOORE for improving its connectivity and robustness, especially if its order is close to that of its predecessor Kautz digraph.

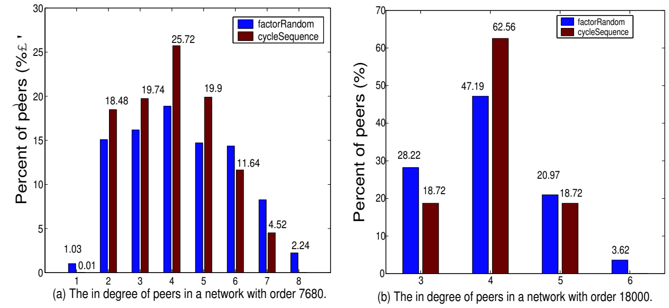

Fig. 2. The in-degree distribution of $IK(4, 7680)$ and $IK(4, 18000)$.

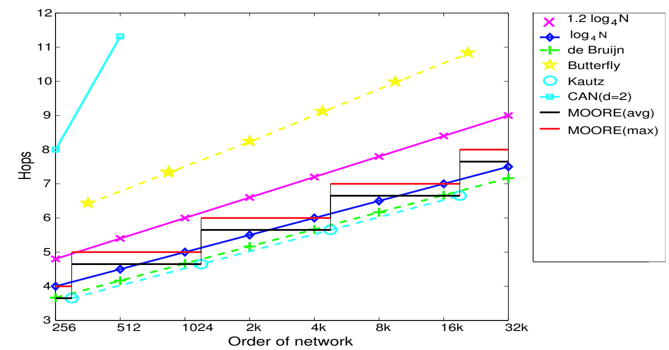

Fig. 3. The average path length of several constant degree topologies.

We know that the order of $IK(4, 7680)$ and $IK(4, 18000)$ is covered by ranges $(n_0, 2n_0]$ and $[3n_0, 4n_0]$, where n_0 denotes the order of $K(4, 5)$ and $4n_0$ equals that of $K(4, 6)$. Thus, the least in-degree of $IK(4, 7680)$ and $IK(4, 18000)$ is 1 and 3 according to Theorem 3, as shown in Figure 2. Furthermore, the in-degree of most nodes is around d and that of few nodes is around the tail of its in-degree distribution figure, if the order of MOORE is adjacent to any multiple of n_0.

B. Diameter and path length distribution of MOORE

Corollary 3: Given a MOORE with arbitrary order N and out-degree d, its diameter is $D_l = \lceil \log_d(N) - \log_d(1 + 1/d)\rceil$.

Proof: First, let's calculate D such that $d^{D-2}(d+1) < N < d^{D-1}(d+1)$. Thus, the length of node identifier must be D, and we can always find a pair of vertices at distance D. Thus $D_l = \lceil \log_d(N) - \log_d(1 + 1/d)\rceil$. ■

According to (2), this is the smallest diameter for any number of vertices N, $d^{D-1} + d^{D-2} \le N \le d^D + d^{D-1}$, and solves the *order/diameter* problem. A lookup for resource or node initiated by any node can reach its destination in $O(\log_d N)$ hops, and the same result holds for the resource's publishing operation.

We evaluate the average path length of MOORE in different scales (from 256 peers up to $32K$ peers) and compare it with other constant degree digraphs with the same degree 4, such as CAN with $d = 2$, 4-dimensional butterfly, de Bruijn, and Kautz. In each experiment, we sample at least $N' = \lceil N/2\rceil$ nodes randomly, and let each sampled node launch a routing to other $N - 1$ nodes, then analyze the average path length over $N' \times (N - 1)$ routings.

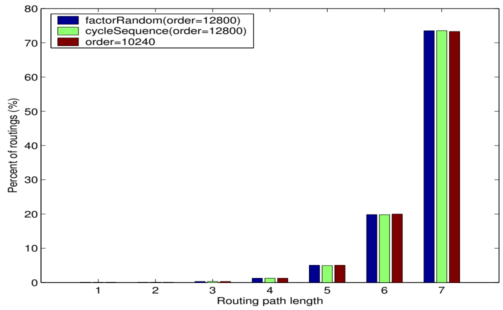

Fig. 4. The path length distribution of $IK(4, 12800)$ and $IK(4, 10240)$.

The simulation result is shown in Figure 3. The curves of the average path length of butterfly, de Bruijn and Kautz are dashed lines because their orders are discrete sequences, as opposed to continuous ranges. The average path and diameter of MOORE are denoted as MOORE(avg) and MOORE(max), respectively, and their curves are solid lines because of their arbitrary order size. MOORE(avg) and MOORE(max) are only a little more than $\log_4 N$ at partial points of *order* axis, but less than the curve of $1.2 \times \log_4 N$, butterfly and CAN at whole *order* axis. We do not compare MOORE with k-dimensional CCC directly in Figure 3 because the degree of CCC is only 3, but the average path length and diameter of MOORE with out-degree 3 is less than that of CCC in reality. Furthermore, the average path length of MOORE with different scales is trivially different if the scales are covered by identical range.

Corollary 4: With the shortest path routing scheme, MOORE can achieve low congestion.

Proof: Figure 4 shows the distribution of routing path lengths of $IK(4, 12800)$ and $IK(4, 10240)$, more than 90% of routing path length are close to the diameter of MOORE. We also find that there exists a similar result under any scale of MOORE. This is closer to the result of the ideal, long path routing scheme used by [9], [16]. Therefore, it is reasonable that MOORE also can achieve the similar low congestion characteristic discussed by Xu [10] and Li [16], although our algorithm adopts a shortest path routing scheme. ■

Corollary 5: Messages caused by node joining and departing operations are at most $2.5d \log_d N$ and $(2.5d + 1) \log_d N$, and only d and $2d$ nodes need to update routing tables.

Proof: Algorithm 4 must find d out-neighbors in order to construct its routing table, and inform d in-neighbors to update their routing table. Algorithm 5 may need to find a substitute node first. Therefore, the former part of Corollary 5 holds because the routing length is less than $1.2 \log_d N$, and the latter part also holds according to the two algorithms. ■

VII. Conclusion

MOORE is the first effective and practical P2P network based on the incomplete Kautz digraph, and is $O(\log_d N)$ in diameter with constant degree. It constructs an overlay digraph for all network sizes and any constant degree, and achieves optimal diameter, high performance, good connectivity and low congestion. In the future, we will improve MOORE to support more query types such as range and multi-attribute query, and consider the locality of the physical network to reduce latency.

References

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proc. ACM SIGCOMM*, pages 161–172, 2001.

[2] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *IEEE/ACM Trans. Networking*, 11(1):17–32, 2003.

[3] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–350, 2001.

[4] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proc. International Peer-to-Peer Symposium*, pages 53–65, Cambridge, MA, USA, March 2002.

[5] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proc. the 21st ACM PODC*, pages 183–192, Monterey,CA, August 2002.

[6] F. Kaashoek and D. Karger. Koorde: A simple degreeoptimal distributed hash table. In *Proc. International Peer-to-Peer Symposium*, pages 98–107, Berkeley,CA,USA, February 2003.

[7] S. Banerjee and D. Sarkar. hypercube connected rings: a scalable and fault-tolerant logical topology for optical networks. *Computer communications*, 24:1060–1079, 2001.

[8] K. N. Sivarajan and R. Ramaswami. Lightwave networks based on de bruijn graphs. *IEEE/ACM Trans. Networking*, 2(1):70–79, 1994.

[9] G. Panchapakesan and A. Sengupta. On a lightwave networks topology using kautz digraphs. *IEEE Computer*, 48(10):1131–1138, 1999.

[10] J. Xu, A. Kumar, and X. X. Yu. On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. *IEEE J. Select. Areas Commun.*, 22(1):151–163, 2004.

[11] M. Naor and U. Wieder. Novel architecture for p2p applications: the continuous-discrete approach. In *Proc. ACM Symposiumon Parallel Algorithms and Architectures*, pages 50–59, San Diego, California, USA, June 2003.

[12] P. Fraigniaud and P. Gauron. D2B: a de bruijn based content-addressable network. *Theor. Comput. Sci.*, 355(1):65–79, 2006.

[13] P. Fraigniaud and P. Gauron. An overview of the content-addressable network d2b. In *Proc. the 22st ACM PODC*, page 151, Boston,USA, July 2003.

[14] D. Loguinov, J. Casas, and X. Wang. Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. *IEEE/ACM Trans. Networking*, 13(5):1107–1120, October 2005.

[15] A. T. Gai and L. Viennot. Broose: a practical distributed hashtable based on the de bruijn topology. In *Proc. the International Conference on Peer-to-Peer Computing*, pages 167–174, Switzerland, August 2004.

[16] D. Li, X. Lu, and J. Wu. Fissione: A scalable constant degree and low congestion dht scheme based on kautz graphs. In *Proc. IEEE INFOCOM*, pages 1677–1688, Miami,Florida,USA, March 2005.

[17] P. Tvrdik. Kautz necklaces. Technical Report Research Report 94-08, LIP ENSL, 69364 Lyon, France, March 1994.

[18] P. Tvrdik. Partial kautz line digraphs with maximal connectivity. Technical Report Research Report 94-15, LIP ENSL, 69364 Lyon, France, April 1994.

[19] M. Imase and M. Itoh. Design to minimize diameter on building-block network. *IEEE Trans. Computers*, 30(6):439–442, June 1981.

[20] M. Imase and M. Itoh. A design for directed graphs with minimize diameter. *IEEE Trans. Computers*, 32(8):782–784, August 1983.

[21] M. Miller and J.Siran. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics*, 61:1–63, December 2005.

[22] M. A. Fiol and A. S. Llado. The partial line digraph technique in the design of large interconnection networks. *IEEE Trans. Computers*, 41(7):848–857, July 1992.

[23] G. D. Caro, F. Ducatelle, P. Heegaard, M. Jelasity, R. Montemanni, and A. Montresor. Evaluation of basic services in ahn,p2p and grid networks. http://www.cs.unibo.it/bison/deliverables/D07.pdf, February 2005.