Quasi-Kautz Digraphs for Peer-to-Peer Networks

Deke Guo, *Member, IEEE,* Jie Wu, *Fellow, IEEE,* Yunhao Liu, *Senior Member, IEEE,* Hai Jin, *Senior Member, IEEE,* Hanhua Chen, *Member, IEEE,* and Tao Chen, *Member, IEEE*

Abstract—The topological properties of peer-to-peer (P2P) 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. The Kautz digraph is one of these topologies that have many desirable properties. Unlike interconnection networks, peer-to-peer networks need a topology with an arbitrary order and degree, but the 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 quasi-Kautz digraph with $O(\log_d n)$ diameter and constant degree under a dynamic environment. The diameter and average routing path length, respectively, are shorter than that of CAN, butterfly, and cube-connected-cycle, and are close to that of the 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.

Index Terms—Constant degree networks, Kautz digraphs, peer-to-peer networks

1 Introduction

Structured peer-to-peer (P2P) networks have 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. They impose a certain topology structure on the overlay network and control the placement of data, thus exhibiting several unique properties that unstructured P2P networks lack. In general, the topological properties of structured P2P networks are critical factors that dominate the performance of these systems. The most common concerns about topological properties are peer degree and network diameter. The degree of a peer denotes the number of overlay connections attached to it. The diameter indicates the largest number of hops that must be traversed in order to transmit a message between any two peers in the worst case.

Several non-constant and constant degree interconnection networks have been used as the ideal topology of structured P2P networks. The degree and diameter increase logarithmically with respect to the order of the network for non-constant degree interconnection networks, such as hypercube [7] and ring digraph. The diameter increases logarithmically with respect to the order of the network, whereas the degree of each node remains fixed, regardless of the order of the network, for constant degree interconnection networks, such as cube-connected-cycle [8] (CCC), butterfly [5], d-dimensional torus [7], de Bruijn [9], and Kautz digraph [10]. Among existing structured P2P networks, Chord [2], Pastry [3], Tapestry [11], and Kademlia [4] are based on the hypercube topology, Viceroy [5] and Ulysses [12] are based on the butterfly topology [13], Cycloid [14] is based on the CCC topology, CAN [1] is based on the d-dimensional torus topology, Koorde [6], Distance Halving [15], D2B [16], ODRI [17] and Broose [18] are based on the de Bruijn topology, and FissionE [19] is based on the Kautz topology.

The degree of a node in the Butterfly network is four, whereas that in Ulysses is $O(\log n)$. The degree of a node in Viceroy or Cycloid is seven and cannot be a general constant integer. The expected degree of a node in D2B is constant, but its high probability bound is $O(\log n)$, ie., some peers would be of degree $O(\log n)$. Koorde and distance-halving embed a de Bruijn network on a ring, and employ equivalent connection rules. The only difference is that the node degree of distance-halving must be two, whereas that of Koorde can be an arbitrary integer. ODRI is another scheme based on the de Bruijn network, whereas the details are still under investigation. Broose is a de Bruijn version of Kademlia that was proposed to increase the reliability of de Bruijn based structured P2P networks. Among the known structured P2P networks, only the degree of a node in CAN and Koorde definitely remains fixed, and can be an arbitrary integer.

In the design of structured P2P networks, there are two important requirements. First, P2P networks always pursue a topology with arbitrary order and degree in order to deal with the uncontrolled dynamic operations of nodes, such as joining, departing and failing. Second,

- *D. Guo and T. Chen are with the Key laboratory of Science and Technology for C^{4}ISR Technology, School of Information Systems and Management, National University of Defense Technology, Changsha 410073, P.R. China. E-mail: {guodeke,emilchenn}@gmail.com.*
- *J. Wu is with the Department of Computer and Information Sciences, Temple University, 1805 N. Borad Street, Philadelphia, PA 19122. E-mail: jiewu@temple.edu.*
- *Y. Liu is with the Computer Science Department, Hong Kong University of Science and Technology, Hong Kong. E-mail: liu@cse.ust.hk.*
- *H. Jin and H. Chen are with School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, P.R. China. E-mail:{hjin,chenhanhua}@hust.edu.cn*

P2P networks attempt to design a topology with the smallest diameter given n nodes and fixed degree d since reducing the diameter can improve the performance of structured P2P networks due to the following fact. The P2P networks are overlay networks, in which one hop transmission usually traverses many links and devices in the underlying physical networks and consequently has non-trivial overhead of delay and traffic.

It is well known that constant degree interconnection networks can satisfy the second requirement, and the Kautz digraph obtains the smallest diameter compared to others. The reason is that the Kautz digraph almost achieves the *Moore bound* [20], the order n of a digraph with maximum out-degree d and diameter D meets the constraint: $n \leq (d^{D+1}-1)/(d-1)$ (with more details in Section 2). Unfortunately, constant degree interconnection networks impose an inherent constraint on the number of vertices they can support. For example, the order of a Kautz digraph must be $d^{D-1}(d+1)$ for a given degree d and any value of diameter D. In other words, it can be one of a series of discrete integers, but cannot cover all possible integers. The Kautz digraph therefore cannot satisfy the first requirement, and cannot be directly used to design a structured P2P network. Although the generalized Kautz digraph extends the Kautz digraph for a general number of vertices, it is required to reconstruct the whole topology once the number of vertices changes [21], [22]. Due to the frequent changes of peers in P2P networks, the generalized Kautz digraph is also not suitable for structured P2P networks.

In this paper, we design a quasi-Kautz digraph with an arbitrary network order and node degree which can satisfy the above two requirements and still retain the key properties of a Kautz digraph. We then propose MOORE: the first effective and practical P2P network based on the quasi-Kautz digraph with $O(\log_d n)$ diameter and constant degree under a dynamic environment. The diameter and average routing path are $\lceil \log_d \frac{n}{d+1} + 1 \rceil$ and $\log_d n$, respectively. They are shorter than that of CAN, butterfly, and CCC, but close to that of the de Bruijn and Kautz digraphs. The message costs of node joining and departing operations are at most $2.5d\log_d n$ and $(2.5d+1)\log_d n$, respectively. MOORE can achieve optimal diameter, high performance, good connectivity, and low congestion.

The main contributions of this paper are as follows:

1) We present the definition, construction procedure and theoretical results of a quasi-Kautz digraph with arbitrary order and node degree. It satisfies the two important requirements and retains desirable properties of a Kautz digraph, such as optimal diameter, constant out-degree, simple routing scheme and low congestion.
2) We design a novel structured peer-to-peer network based on the quasi-Kautz digraph, and a suitable resource distribution policy, production methods of resource and node identifier, and a shortest path routing scheme.
3) We propose some essential algorithms to handle the dynamic operations of nodes, such as node joining and departing, and network expanding and shrinking. These algorithms can preserve the desirable structure of the 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 2 surveys the definition and emulation methods of the Kautz digraph. Section 3 proposes the theory of a quasi-Kautz digraph and its construction procedure. Section 4 describes the detailed design of MOORE. Section 5 presents strategies to expand and shrink the entire topology. Section 6 analyzes and evaluates the characteristics of MOORE. The conclusions and future work are discussed in Section 7.

2 Related work

2.1 Kautz digraph

The topology of a structured P2P network is usually modeled by a graph or digraph in which vertices stand for nodes while edges represent overlay connections. Many efforts have been made to address the *degree/diameter* problem, which determines the largest graphs or digraphs of given maximum degree and given diameter. The order n of a digraph with maximum out-degree d and diameter D is not larger than a general *Moore bound* [20], [23] as follows:

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

Many research activities related to the degree/ diameter problem have proved that non-existence of digraphs achieve the general upper bound for the parameters $d \geq 3$ and $D \geq 3$ [24]. The best lower bound on the order of digraphs of maximum out-degree d and diameter D is as follows: For maximum out-degree d=2 and diameter $D \geq 4$, $n \geq 25 \times 2^{D-4}$. For the remaining values of maximum d and diameter D, a general lower bound is $n \geq d^D + d^{D-1}$ [20]. Among existing non-trivial digraphs, this best lower bound is only obtained by Kautz digraphs defined using either an alphabet (the standard method) or congruent arithmetic [25] as follows:

Definition using an 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 $E(K(d,D)) = \{\langle x_1x_2...x_D, x_2, ...x_D\alpha\rangle \,|\, \alpha \in Z_d, \alpha \neq x_D\}$. Figure 1 plots an example of Kautz$(2,2)$.

Definition using congruent arithmetic [21], [22]: Let $GK(d,n)$ denote a generalized Kautz digraph with degree d and order n, respectively. The vertex set and arc set of the generalized Kautz digraph are denoted as $V(GK(d,n)) = \{0,...,n-1\}$ and $E(GK(d,n)) = \{\langle i, (-d \times i - \alpha) \bmod n\rangle \mid 1 \leq \alpha \leq d\}$.

Besides the degree/diameter problem, structured P2P networks also focus on the *order/degree* problem, which determines the smallest diameter in a digraph of order n and maximum out-degree d. Based on the Moore bound of the degree/diameter problem, a lower bound of the order/degree problem can be derived as

$$D \geq \lceil \log_d (n(d-1)+1) \rceil - 1.$$

In practice, all existing digraphs cannot achieve this lower bound for the parameters $d \geq 3$ and $D \geq 3$ [24]. The best upper bound on the diameter of digraphs of maximum out-degree d and order n is $\lceil \log_d \frac{n}{d+1} + 1 \rceil$. Among all existing non-trivial digraphs, the best upper bound is only possessed by the Kautz digraph.

2.2 Emulation of Kautz digraph

The topology is incrementally extendable if its definition allows graphs of arbitrary order 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 [25]. The fundamental reason is that the generalized Kautz digraph requires reconstruction of the whole topology once the number of vertices changes.

The most related research work revolves around FISSIONE, which uses a Kautz graph $K(2,D)$ as its static topology and proposes some emulation methods of $K(2,D)$ to deal with the dynamic operations of nodes. It, however, cannot support Kautz digraphs with arbitrary degree, except degree 2, and suffers from poor lookup performance and weak connectivity since the degree of each peer is too small. Furthermore, the emulation methods of $K(2,D)$ are not suitable to a general Kautz graph $K(d,D)$ where $d>2$. Thus, FISSIONE is not incrementally extendable.

MOORE attains the best upper bound of the order/degree problem mentioned above. Even the order is an arbitrary value. However, it only works well under a relative static or moderately dynamic environment, and suffers from low robustness in highly dynamic environments due to maintaining topology. To address these issues, we improved MOORE by introducing another structured P2P network based on a balanced Kautz tree and Kautz ring in [26]. Recently, Zhang et al. reconsidered the design problem of structured P2P networks mentioned in this work, and also employed a linear digraph to emulate the Kautz digraph [27]. They adopted a fully distributed manner to maintain the node identifier space at the cost of high overhead, while MOORE prefers centralized servers.

3 Quasi-Kautz Digraph

3.1 Definition of quasi-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 $\Gamma_G^-(u)=\{v \in V(G) \mid \langle v,u\rangle \in E(G)\}$, 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 $\Gamma_G^+(u)=\{v \in V(G) \mid \langle u,v\rangle \in E(G)\}$, and $\delta_G^+(u)=|\Gamma_G^+(u)|$ is the out-degree of vertex u.

Given a Kautz digraph $K(d,D)$, we construct an arc set $E' \in E(K(d,D))$ such that each vertex of $K(d,D)$ appears as the head and tail of at least one arc of E', where $|E'|=n$ and $d^D+d^{D-1}<n<d^{D+1}+d^D$.

Definition 1: A digraph of fixed out-degree d and order n, $IK(d,n)$, is a *quasi-Kautz digraph* if:

1) $IK(d,n)$ has arcs of E' as vertices.
2) For each arc (u,v) in E', check the following: For each w in (v,w) in E, if $(v,w) \in E'$, then add an α-arc from vertex (u,v) to vertex (v,w) in $IK(d,n)$; otherwise, select z such that $(z,w) \in E'$, then add an β-arc from vertex (u,v) to vertex (z,w) in $IK(d,n)$.

The Kautz digraphs $K(d,D)$ and $K(d,D{+}1)$ are called the predecessor and successor Kautz digraph of $IK(d,n)$, respectively. According to Definition 1, each arc $\langle u,v\rangle$ in E' can be denoted as a vertex labeled $uv=u_1u_2u_Dv_D$ of $IK(d,n)$ where $u_2u_3...u_D$ equals to $v_1v_2...v_{D-1}$. In this paper, we will not distinguish strictly between an arc of $K(d,D)$ and its corresponding vertex in $IK(d,n)$. In other words, we may use $\langle u,v\rangle$ to denote a vertex of $IK(d,n)$. It is clear that the out-degree of any vertex of $IK(d,n)$ is d. Note that the method used to choose z from multiple candidates will be discussed in Section 4.1.

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

1) Discover the largest Kautz digraph $K(d,D)$ satisfying that $d^D + d^{D+1} < n$.
2) Construct a subset E' of $E(K(d,D))$ such that $E' = n$ and the constraint on E' mentioned above is satisfied.
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 mentioned in Definition 1.

The general procedure can result in different quasi-Kautz digraphs, with the same number of vertices, due to a different arc set E'. The procedure ensures that the minimum *in-degree* of nodes in the resulting quasi-Kautz digraph is not less than 1. It alone, however, is not enough to ensure that the quasi-Kautz digraph can inherit desirable properties of the Kautz digraph. Therefore, a method for careful selection of the arc set E' is necessary.

[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 $IK(d,n)$.

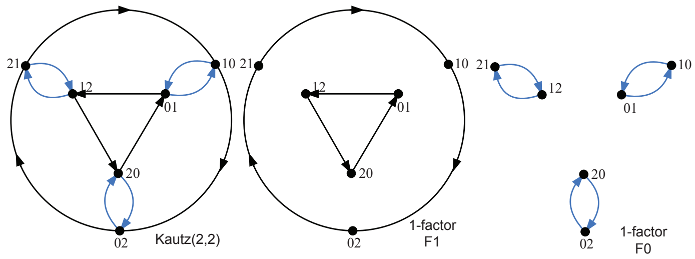


Fig. 1. 1-factorization of a Kautz digraph $K(2,2)$.

3.2 Construction of quasi-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. Theorem 1 proves that the Kautz digraph has a 1-factorization, which will be used to derive a special construction procedure of the quasi-Kautz digraph. Before in-depth analysis, we first introduce several definitions as follows:

Definition 2: Let $Lshift$ denote a binary operation such that $Lshift(x_1...x_{D-1}x_D,i) = x_1...x_{D-1}x'_D$, where $0 \le i \le d{-}1$. 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, $x'_D = (x_D + i + 1) \bmod (d+1)$ [25].

Definition 3: Let $Rshift$ denote a binary operation such that $Rshift(x_1x_2...x_{D-1}x_D,i){=}x'_1x_2...x_{D-1}x_D$, where $0{\le}i{\le}d{-}1$. 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 any vertex $x{=}x_1x_2...x_D$ in $K(d,D)$ and $0{\le}i{\le}d{-}1$, the left k-shift operation and right k-shift operation, denoted as σ^i_k and σ^{-i}_k, respectively, are defined as follows:

$$\sigma^i_1(x) = \begin{cases} Lshift(x_2...x_Dx_1,i), \text{ if } x_1 \neq x_D \\ Lshift(x_2...x_Dx_2,i), \text{ if } x_1 = x_D \end{cases} \tag{2}$$

$$\sigma^i_k = \sigma^i_{k-1}(\sigma^i_1) \tag{3}$$

$$\sigma^{-i}_1(x) = \begin{cases} Rshift(x_Dx_1...x_{D-1},i), \text{ if } x_1 \neq x_D \\ Rshift(x_{D-1}x_1...x_{D-1},i), \text{ if } x_1 = x_D \end{cases} \tag{4}$$

$$\sigma^{-i}_k = \sigma^{-i}_{k-1}(\sigma^{-i}_1). \tag{5}$$

For any vertex x, vertices $\sigma^i_1(x)$ and $\sigma^{-i}_1(x)$ 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 and σ^{-i}_1 operations.

Theorem 1: The arc set $E(K(d,D))$ can be partitioned into d arc-disjoint 1-factors $\mathcal{F}^0,...,\mathcal{F}^{d-1}$ under the corresponding left 1-shift operation σ^i_1 $(0{\le}i{\le}d{-}1)$. That is, $K(d,D)$ 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 the left 1-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.

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 $j$
```

As discussed above, each vertex of $K(d,D)$ is covered by at least one cycle under the operation σ^i_1. Let us suppose that 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 the 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 $\mathcal{F}^i$ of $K(d,D)$. Furthermore, for any vertex x of $K(d,D)$, the arc covering it is different in different 1-factors. Therefore, those 1-factors are mutually arc-disjoint, and $K(d,D)$ has a factorization. Therefore, Theorem 1 holds. □

As shown in Figure 1, all arcs of a Kautz digraph $K(2,2)$ can be partitioned into two arc-disjoint 1-factors. The Kautz digraph $K(2,2)$ therefore has a 1-factorization. According to Definition 1, the corresponding arc of each vertex $x{=}x_1...x_Dx_{D+1}$ of a $IK(d,n)$ is contained by a unique 1-factor in the predecessor Kautz digraph of the $IK(d,n)$. The identifer or label of that 1-factor can be calculated by $\mathcal{F}(x){=}Distance\,(\sigma^0_1(x_1x_2...x_D), x_2x_3...x_{D+1})$, where the function $Distance$ is given by Algorithm 1.

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

Proof: We know that each vertex x of $K(d,D)$ is covered by an arc $\langle x,\sigma^i_1(x)\rangle$ in 1-factor $\mathcal{F}^i$ where $0 \le i < d$. According to Definition 1, the vertex labeled $\langle x,\sigma^i_1(x)\rangle$ is incident on d vertices in a $IK(d,d^D + d^{D-1})$ induced by a 1-factor $\mathcal{F}^i$. This proves that the quasi-Kautz digraph induced by $\mathcal{F}^i$ is d-out-regular.

There is an α-arc from vertex $\langle\sigma_1^{-i}, x\rangle$ to vertex $\langle x, \sigma_1^i(x)\rangle$ in a quasi-Kautz digraph induced by a 1-factor $\mathcal{F}^i$. Furthermore, the arc from vertex $\sigma_1^{-j}(\sigma_1^i(x))$ to vertex $\sigma_1^i(x)$ is not in the $\mathcal{F}^i$ where $0{\leq}j{\leq}d-1$ and $j{\neq}i$. According to the proof of Theorem 1, we know that there exists an arc $\langle\sigma_1^{-i}(\sigma_1^{-j}(\sigma_1^i(x))), \sigma_1^{-j}(\sigma_1^i(x))\rangle$ in the $\mathcal{F}^i$. Thus, there exists $d{-}1$ β-arcs from vertices $\langle\sigma_1^{-i}(\sigma_1^{-j}(\sigma_1^i(x))), \sigma_1^{-j}(\sigma_1^i(x))\rangle$ to vertex $\langle x, \sigma_1^i(x)\rangle$. In summary, each vertex $\langle x, \sigma_1^i(x)\rangle$ have d number of in-neighbors, and the quasi-Kautz digraph induced by the 1-factor $\mathcal{F}^i$ therefore is a d-in-regular and d-regular digraph.

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

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

1) In order to construct a $IK(d, n)$ where $k(d^D + d^{D-1}) \leq n \leq (k+1)(d^D + d^{D-1})$, we start with a d-regular quasi-Kautz digraph $IK(d, d^D + d^{D-1})$ induced by the 1-factor $\mathcal{F}^0$ of $K(d, D)$ through Algorithm 5. The $K(d, D)$ can be achieved from an initial Kautz digraph by invoking this procedure repeatedly.
2) We add vertices corresponding to all arcs of $k-1$ 1-factors $\mathcal{F}^1, \mathcal{F}^2, ..., \mathcal{F}^{k-1}$ to the d-regular digraph produced in the first step, and then achieve a new d-regular digraph $IK(d, k(d^D + d^{D-1}))$ by using Algorithm 3 recursively.
3) We then add vertices corresponding to $n - k(d^D + d^{D-1})$ arcs, denoted $\mathcal{F}^{k'}$, of another 1-factor $\mathcal{F}^k$ to the new d-regular digraph by using Algorithm 3 recursively.

Note that Theorem 2 guarantees the correctness of the first step. The last step is based on proper choice of the added arcs as discussed in Section 4. In order to achieve higher connectivity, the arc selection polices must make the minimum in-degree of the final digraph as large as possible. Theorem 3 shows the low and upper bounds on the minimum in-degree of a resulting $IK(d, n)$.

Theorem 3: Given any value of n, any quasi-Kautz digraph $IK(d, n)$ always holds that $k{\leq}\delta^-(IK(d, n)){\leq}d$ where $k(d^D{+}d^{D-1}){\leq}n{\leq}(k{+}1)(d^D + d^{D-1})$ and $1{\leq}k{<}d$.

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 $\mathcal{F}^0, \mathcal{F}^1..., \mathcal{F}^k$, but the result is the same for any $k+1$ 1-factors. The special construction procedure can produce the needed quasi-Kautz digraph mentioned in this theorem. Theorem 2 can also guarantee that the quasi-Kautz digraph induced by any k 1-factors of $K(d, D)$ is a d-regular digraph.

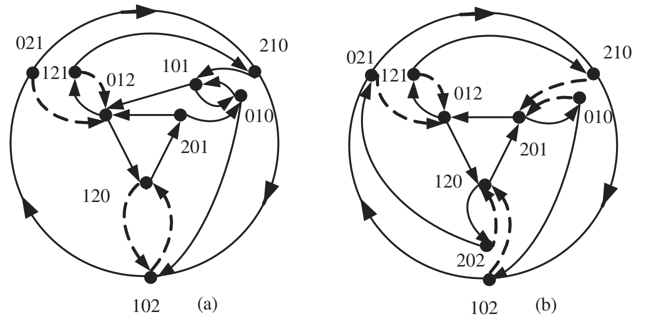


Fig. 2. Two different shapes of a quasi-Kautz digraph $IK(2, 9)$.

Adding any vertex x induced by $\mathcal{F}^{k'}$ has an effect on one out-arc of at most d existing nodes. Node x needs to inform its $(i{+}1)^{th}$ predecessor to update the $(i{+}1)^{th}$ out-arc (a β arc) with a new α-out-arc incident on node x where $0{\leq}i{\leq}k{-}1$. As a result, the in-degree of the node at the other end of the original $(i+1)^{th}$ out-arc of the $(i+1)^{th}$ predecessor of vertex x decreases by one. If the arc corresponding to its $(k+1)^{th}$ predecessor has been added previously, node x also informs this predecessor to add an α-arc to itself. For $k+1{\leq}i{\leq}d{-}1$, other $d{-}k{-}1$ predecessors of node x are induced by 1-factors $\mathcal{F}^i$ and do not exist in $IK(d, n)$. There, however, exists a β arc from a node corresponding to an arc $\langle\sigma_1^{-k}(\sigma_1^{-i}(x_2...x_{D+1})), \sigma_1^{-i}(x_2...x_{D+1})\rangle$ to the node x if that arc is in $\mathcal{F}^{k'}$.

According to the above analysis, the in-degree of vertices induced by $\mathcal{F}^{k'}$ is at least k, but less than d, except when $k{=}d-1$ and arcs of $\mathcal{F}^i$ form cycles. The in-degree of vertices induced by previous k 1-factors should not be less than $d-1$, and can reach d in some scenarios such as in Figure 2 (b). Thus, $k \leq \delta^-(IK(d, n)) \leq d$, and Theorem 3 holds. □

4 MOORE DESIGN

We propose the following strategies to organize peers into an efficient overlay network which can guarantee the logarithmic network diameter and constant out-degree of each peer. First, each peer obtains a logical identifier from an identifier space, and uses its IP address as a physical identifier. Second, each peer maintains d neighbor peers according to a topology rule. Third, any resource gets an identifier from an identifier space which contains the identifier space of peers. Resources are distributed to given peers based on the longest prefix matching rule. Based on the above three strategies, we propose a routing scheme to support different operations effectively, such as resource distribution, resource querying and topology maintenance.

MOORE uses the quasi-Kautz digraph as its topology structure, which evolves from an initial Kautz digraph in a distributed manner. The Kautz digraph can be constructed through many mature centralized methods, so we do not consider related details in this paper. In practice, MOORE needs to deal with the following dynamic operations: topology expanding, topology shrinking, node joining and departing. It is these operations that drive the evolution of MOORE. We will first propose essential algorithms to implement the two dynamic operations of peers in this section, and then explain how to expand and shrink the MOORE topology corresponding to a Kautz digraph in Section 5.

4.1 Overview

The quasi-Kautz digraph inherits many desirable characteristics of the Kautz digraph, and is more practical than the Kautz digraph because its order can be of an arbitrary order. Therefore, MOORE selects the quasi-Kautz digraph as its topology in a dynamic environment. There is an injection mapping from nodes in MOORE to vertices in a corresponding quasi-Kautz digraph. The topology of MOORE evolves from an initial Kautz digraph through dynamic operations of nodes and must always satisfy the constraints mentioned in Definition 1.

As mentioned in the latter, the i^{th} out-neighbor of an existing node $x{=}x_1x_2...x_D$ is $\langle x_2x_3...x_D, \sigma_1^i(x_2x_3...x_D)\rangle$. In practice, the i^{th} desired out-neighbor might not appear in MOORE. In this situation, node x must select a substitute for its i^{th} desired out-neighbor from at most d existing nodes labeled $\langle\sigma_1^{-j}(\sigma_1^i(x_2x_3...x_D)), \sigma_1^i(x_2x_3...x_D)\rangle$ where $0{\le}j{<}d$. Recall that Definition 1 does not point out a method to choose the substitute from multiple existing candidates. To deal with this issue, MOORE chooses the node as a substitute labeled $\sigma_1^{-F(x)}(\sigma_1^i(x_2x_3...x_D)), \sigma_1^i(x_2x_3...x_D)\rangle$ if it exists. Otherwise, MOORE chooses one randomly from those candidates.

For any resource to be distributed in MOORE, it is assigned a long d-ary identifier $x{=}x_lx_2...x_l$ according to its value of single or multiple dimension attributes. 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 a resource identifier should be larger than that of a node identifier. If we fix the out-degree of each node in MOORE, 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 in MOORE, respectively.

Assume the successor Kautz digraph of $IK(d,n)$ is $K(d,D)$, a resource labeled $x_1x_2...x_D...x_l$ is stored and maintained by its preferred host labeled $x_1x_2...x_D$ if this node exists in $IK(d,n)$. Otherwise, the resource will be taken over by its second host labeled $\langle x_1x_2...x_{D-1}, \sigma_1^s(x_1x_2...x_{D-1})\rangle$ in $IK(d,n)$. In the remainder of this paper, let s denote the identifier of the 1-factor that was selected to induce the quasi-Kautz digraph with the same order as $K(d, D-1)$. MOORE can ensure that at least the second host of each resource appears in MOORE. In general, the default value of s is 0, and the second host of resource x is labeled $x_1x_2...x_{D-1}x_1$ (if $x_1{\neq}x_{D-1}$) or $x_1x_2...x_{D-1}x_1{+}1$ (if $x_1{=}x_{D-1}$). For example, it is the node 210 that stores a resource labeled 212120212 when the node 212 does not appear in MOORE, as shown in Figure 2.

4.2 Mapping resources onto resources' identifier space

Each resource accessible through MOORE will receive an identifier from the identifier space Z_d^l. Different resources are allowed to receive the same identifier. The mapping of resources onto Z_d^l can be implemented in several ways. Literature [19] proposed a determinate algorithm to generate an identifier with two as a base for each resource. In reality, the base of a quasi-Kautz digraph used by MOORE is often larger than two for the sake of decreasing its diameter and improving its connectivity. Therefore, this paper considers another $Kautz\_hash$ algorithm to generate an identifier with any base for each resource. The $Kautz\_hash$ uses three parameters: key denotes the original identifier of the 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 produces a long binary string by hashing the key according to a given consistent hash function, for example $SHA-1$. 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.

4.3 Mapping nodes 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 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 will be better if we start with an initial identifier space and then enlarge or shorten it with the increase or decrease of the number of nodes, respectively. Let $Z_d^{m_0}$ denote the initial identifier space where $m_0 < m$. 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 MOORE, the initial identifier space should be extended to $Z_d^{m_0+1}$ so as to 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 $\mathcal{F}^0$ of $K(d,m_0)$, then the initial nodes form another d-regular quasi-Kautz digraph $IK(d,d^{m_0}+d^{m_0-1})$ according to Algorithm 5. As discussed later, this process does not cause additional overhead except $d^{m_0}+d^{m_0-1}$ messages to start the process. In order to maintain better topological 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 quasi-Kautz digraph mentioned above. Any arc choice policy first takes the arcs of the second 1-factor $\mathcal{F}^1$, then takes the arcs of the third 1-factor $\mathcal{F}^2$, 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 [25] suggests to take arcs of one cycle in each 1-factor, then arcs of another cycle, and so on. The random choice policy, denoted as $factorRandom$, selects arcs randomly from a given 1-factor. The difference between these two policies is that the former can make the in-degree of more new vertices reach $k{+}1$. The n denotes the number of existing nodes in MOORE, and k satisfies that $k(d^{m_0}+d^{m_0-1}) \leq n \leq (k{+}1)(d^{m_0}{+}d^{m_0-1})$. 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 [25]. The reason is that the $(k+1)^{th}$ predecessor of a newly added arc has been added previously unless it is the first selected arc of a cycle.

Recall that the in-degree of at most k nodes induced by previous k 1-factors decreases by one once a new node x joins MOORE. Here, the $(k+1)^{th}$-out-arc of existing peer $\sigma_1^{-i}(x)$ incidents on one of those k nodes, where $0 \leq i \leq k{-}1$. As shown in Figure 2(a), the original β-out-arc from vertex 012 to 021 will be updated with an α-out-arc from vertex 012 to 121 once a vertex 121 participates $IK(d,n)$. 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 the previous node's adding operation, and reestablish its $(k+1)^{th}$ in-arc with an 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 topological properties if it combines the policies *inDegreePreserved* and *cycleSequence*.

On the other hand, an identifier allocated to a node may become free 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 $\mathcal{F}^i$ 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.

Algorithm 2 Route(y, message, scheme)

```
Require: Identifier $y$ is not less than $x$
z ← y
if the length of $y$ is larger than $D$ then
    $y \leftarrow y_1y_2...y_D$
if $x = y$ or $x_1x_2...x_{D-1} = y_1y_2...y_{D-1}$ 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 $\langle x_1x_2...x_{D-1}, \sigma_1^s(x_1x_2...x_{D-1})\rangle$
else
    return $result$
```

4.4 Routing scheme

In order to route messages to destinations correctly, each node x must establish links with selected neighbors and construct a routing table when it joins MOORE using Algorithm 3. In addition, each node should 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, or initiate an *insert* message to distribute its resource with identifier y to a responsible node. We propose Algorithm 2 to route those kinds of messages to their destinations along the shortest paths.

Fiol proposed a method to achieve a short path from x to y in [28]: 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 due to the existence of β-out-arcs. As an example, node 021 needs to route to node 012 along the short path $021 \rightarrow 210 \rightarrow 101 \rightarrow 012$, as shown in Figure 2(a). The

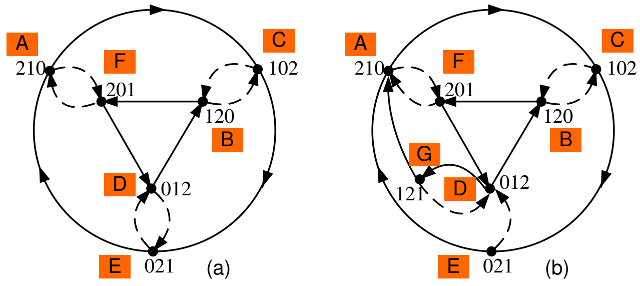


Fig. 3. The topology of MOORE before and after adding a peer 121.

shortest path, however, should be $021 \rightarrow 012$, 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. As shown in our simulation results, Algorithm 2 can achieve low congestion as the long path routing scheme does [10], [19].

Algorithm 2 uses three parameters: y denotes the identifier of a aimed 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 the quasi-Kautz digraph is different from that of the Kautz digraph, 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 an aimed resource, it will forward the message to another destination node defined by the resource distribution policy mentioned above.

4.5 Node joining

To ensure that our routing scheme executes correctly after a new peer participates MOORE, all routing entries of each peer must keep up to date. MOORE handles this issue by a series of local operations that each new peer runs when it joins. The joining procedure includes receiving a node identifier, redistributing resources, and updating routing tables. These operations can be implemented by Algorithm 3.

As for most P2P networks, we assume there are some existing nodes as *entry points* of MOORE, which can receive and process the node joining message. Let $y = y_1y_2...y_{D+1}$ denote an *entry point* of MOORE. Before participating MOORE, a new peer consults node y for its logical identifier $x = x_1x_2...x_{D+1}$ and the identifier k of a current 1-factor according to the management policy of *nodes'*identifier space. In reality, there exists at least two cases of node joining operations. The first case is $\mathcal{F}(x) = k$, which means that the new node belongs to the current 1-factor $\mathcal{F}^k$. The second case is $\mathcal{F}(x) < k$, which means that the new node belongs to the previous 1-factor and a node with the same identifier has joined MOORE, but failed or departed.

Algorithm 3 Node joins(x,y,k)

```
$k \leftarrow \mathcal{F}(x)$
for $i = 0$ to $d$ do
    if $i \leq k$ then
        Node $y$ finds the $address$ of node labeled $z$. Then node $x$ adds $\langle z, address, \alpha \rangle$ as its $(i+1)^{th}$ routing entry, and establishes a link to this node, 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 $\langle \sigma_1^{-j}(\sigma_1^i(x_2x_3...x_{D+1})), \sigma_1^i(x_2x_3...x_{D+1}) \rangle$. The random integer $j$ satisfies that $0 \leq j < k$ and 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 $i \leq k$ then
        $w \leftarrow \langle \sigma_1^{-i}(x_1x_2...x_D), x_1x_2...x_D \rangle$
    else
        $w \leftarrow \langle \sigma_1^{-k}(\sigma_1^{-i}(x_2...x_{D+1})), \sigma_1^{-i}(x_2...x_{D+1})$
    Node $w$ updates one original $\beta$ link with an $\alpha$ or $\beta$ link incident on node $x$, then updates its routing table.
Node $x$ gets resources satisfied that $x$ is their prefix of identifier from node $\langle x_1x_2...x_D, \sigma_1^s(x_1x_2...x_D) \rangle$.
```

In both cases, node x needs to find its successors for establishing out-links and a routing table, then inform at most d existing predecessors to update their links and routing tables, and finally take over its responsible resources from an existing node. The details have been proposed when proving Theorem 3. Given an integer k such that $k(d^D + d^{D-1}) \leq n \leq (k+1)(d^D + d^{D-1})$, we know that 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 except that node x is mapped to the last arc of the current cycle, and its $(k+1)^{th}$ predecessor exists except that node x is mapped to the first arc selected from a cycle. The other j^{th} successor of node x does not exist for $k+1 < j \leq d$, and it needs to find a substitute from nodes belonging to 1-factor $\mathcal{F}^k$, even from nodes belonging to previous 1-factors, in order to keep a constant out-degree. The other j^{th} predecessors of node x also do not exist for $k+1 < j \leq d$. Therefore, node x should find a substitute for its j^{th} predecessor for $k+1 < j \leq d$ from nodes belonging to 1-factor $\mathcal{F}^k$. Node x, however, does not select substitutes for predecessors from nodes belonging to previous 1-factors in order to not increase the in-degree of nodes belonging to previous 1-factors.

The resulting topology of MOORE after adding a new node can be represented pictorially and an example is illustrated in Figure 3. If a node 121 joins MOORE, whose topology is shown in Figure 3 (a), the resulting topology of MOORE is plotted by Figure 3 (b).

4.6 Node departing

The correctness and effectiveness of MOORE relies on the fact that predecessors and successors of each node are up to date. An incorrect neighbor might increase

Algorithm 4 Node departs (x, k)

```
if $\mathcal{F}(x) < k$ then
    $y \leftarrow findSubstitute(x)$
    update($y, k, \mathcal{F}(x)$)
    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 $\langle y_1y_2...y_{m-1}, \sigma_1^s(y_1y_2...y_{m-1})\rangle$ before departing.
    update($x, k, \mathcal{F}(x)$)
update($z, k, l$)
for $i = 0$ to $d$ do
    if $i < k$ 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 $\beta$ 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 $\beta$ 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 $0 \leq j < k$ such that the new destination node exists.
```

the delay of routing a message, and even fail to deliver messages correctly. Therefore, a node departing voluntarily should repair the topology through the following procedures before it leaves.

Let $x = x_1x_2...x_{D+1}$ denote a node departing from MOORE, and k denote the identifier of the current 1-factor. In practice, there exist at least two cases of node departing operations. The first case is $\mathcal{F}(x) = k$, which means that node x belongs to the current 1-factor $\mathcal{F}^k$. $\mathcal{F}(x) < k$ 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 4 can compensate for the negative impact of the node leaving operation. For example, If node 121 departs from MOORE, whose topology is shown in Figure 3 (b), the resulting topology of MOORE is plotted by Figure 3 (a).

In the first case, node x needs to inform its in-neighbors to update the link incident on node x, and transfer its resources to another responsible node defined by the resource distribution policy. In the second case, node x needs to find a node y to replace it, and inform 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 the change of its address. Node y should be selected from nodes belonging to 1-factor $\mathcal{F}^k$, then 1-factor $\mathcal{F}^{k-1}$, and so on. This policy can preserve the desired topology of a backbone subnetwork consisting of nodes belonging to previous 1-factors.

5 TOPOLOGY ADJUSTMENTS

5.1 Problem statements

In general, the topology of MOORE is a quasi-Kautz digraph $IK(d, n)$ where the number of nodes, n, is covered by a unique range $[d^D + d^{D-1}, d^{D+1} + d^D)$. In practice, the topology becomes a Kautz digraph $K(d, D)$ 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 quasi-Kautz digraph whose order equals to the lower boundary of a new range $[d^{D+1} + d^D, d^{D+2} + d^{D+1})$. If the number of nodes reaches $d^{D+2} + d^{D+1}$, a quasi-Kautz digraph becomes a Kautz digraph and is ready to be expanded further.

It is easy to derive a quasi-Kautz digraph $IK(d, d^{D+1} + d^D)$ from its predecessive Kautz digraph $K(d, D)$ by using Definition 1 with the 1-factor $\mathcal{F}^0$ of $K(d, D)$ as the arc set E'. To expand the topology of MOORE similarly, we propose two strategies to update logical identifier of each node and associated algorithms to update out-neighbors and routing tables of each node.

For the first strategy, each node x=$x_1x_2...x_D$ updates its logical identifier with $\langle x, \sigma_1^s(x)\rangle$ such that the new identifier and the original identifier have a common prefix with length D. This strategy is also called the *prefix-preserved expansion strategy*. For the second strategy, each node x updates its logical identifier with $\langle {\sigma_1^{-}}^{s}(x), x\rangle$ such that the new identifier and the original identifier have a common suffix with length D. This strategy is also called the *suffix-preserved expansion strategy*. For the two strategies, existing nodes form the same topology structure. As analyzed later, the two strategies, however, produce different network overhead during the topology expansion process.

The number of nodes in MOORE sometimes decreases to the lower boundary of the range $[d^D + d^{D-1}, d^{D+1} + d^D)$ in practice. In this situation, if some existing nodes want to leave, MOORE needs to shrink its topology to its predecessive Kautz digraph. If the number of nodes in MOORE decreases to $d^{D-1} + d^{D-2}$, the quasi-Kautz digraph might be shrunken further. The shrink operation can be performed by updating the logical identifier, out-neighbors, and routing table of each existing node. There are two possible strategies to update logical identifiers of existing nodes. For the *prefix-preserved shrink strategy*, each existing node x=$x_1x_2...x_{D-1}x_D$ updates its original logical identifier with $x_1x_2...x_{D-1}$ such that the new and original identifiers have a common prefix with length $D - 1$. For the *suffix-preserved shrink strategy*, each node x updates its logical identifier with $x_2x_3...x_D$.

5.2 Prefix-preserved adjustment strategy

The prefix-preserved expansion strategy can be implemented by Algorithm 5. The parameter s in this algorithm denotes the identifier of the 1-factor that was selected to induce the quasi-Kautz digraph with the same order as $K(d, D)$, where the default value of s is 0.

Algorithm 5 Prefix-preserved Expansion ($K(d, D), s$)

```
Require: $K(d, D)$ is a $d$-regular Kautz digraph with diameter $D$.
for each node $x$ labeled $x_1x_2...x_D$ in $K(d, D)$ do
    $x.label \leftarrow \langle x, \sigma_1^s(x) \rangle$
    Node $x$ constructs a temporary routing table.
    for $i = 0$ to $d - 1$ do
        if $s = i$ then
            $z = z_1z_2...z_{D+1} \leftarrow \langle \sigma_1^s(x), \sigma_2^s(x) \rangle$
            $address \leftarrow x.routing[s].address$
            Node $x$ adds $\langle z, address, \alpha \rangle$ as the $(i+1)^{th}$ entry of the temporary routing table.
        else
            $z = z_1z_2...z_{D+1} \leftarrow \langle \sigma_1^{-s}(\sigma_1^i(\sigma_1^s(x)), \sigma_1^i(\sigma_1^s(x)) \rangle$
            $address \leftarrow$ Route($\sigma_1^{-s}(\sigma_1^i(\sigma_1^s(x)), , $node))
            Node $x$ adds $\langle z, address, \beta \rangle$ as the $(i+1)^{th}$ 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.
```

For each node $x = x_1x_2...x_D$, it constructs a temporary routing table by the following operations:

1) Updates its logical identifier with $\langle x, \sigma_1^s(x) \rangle$.
2) Updates the logical identifier of its $(s+1)^{th}$ out-neighbor node $\sigma^s(x)$ with $\langle \sigma_1^s(x), \sigma_2^s(x) \rangle$.
3) Updates the logical identifier of its $(i+1)^{th}$ out-neighbor node $\sigma^i(x)$ with $\langle \sigma_1^{-s}(\sigma_1^i(\sigma_1^s(x)), \sigma_1^i(\sigma_1^s(x)) \rangle$, where $0 \leq i < d$ and $i \neq s$.
4) Discovers the address of a node which updates its logical identifier $\sigma_1^{-s}(\sigma_1^i(\sigma_1^s(x))$ with $\langle \sigma_1^{-s}(\sigma_1^i(\sigma_1^s(x)), \sigma_1^i(\sigma_1^s(x)) \rangle$, where $0 \leq i < d$ and $i \neq s$.

For $0 \leq i < d$ and $i \neq s$, the fresh and original $(i+1)^{th}$ out-neighbors of node x are not the same node, and hence node x must discover the physical address of its new out-neighbor by initiating a query. The new $(s+1)^{th}$ out-neighbor of node x is just the original $(s+1)^{th}$ out-neighbor. Therefore, node x is not necessary to send a query for the physical address of its new $(s+1)^{th}$ out-neighbor. After all existing nodes finish these operations, each of them update its routing table with the temporary routing table, and finally updates links according to its new routing table. As an example, Figure 4 (a) becomes Figure 4 (b) through this algorithm. Theorem 4 proves the network overhead of this type of topology expansion strategy.

Theorem 4: In the case of the prefix-preserved expansion strategy, the expansion of the entire overlay network causes $n \times (d-1)\log_d n$ additional network overhead.

Proof: As mentioned above, each node must explore physical addresses of $d-1$ neighbors by initiating $d-1$ query messages. It is clear that each of these messages will be routed to a destination within at most $\log_d n$ hops. Therefore, the total number of messages caused by expanding the overall topology is at most $n \times (d-1)\log_d n$. Thus, Theorem 4 holds. □

In contrast to expanding the overlay, MOORE shrinks its topology when the number of existing nodes decreases to the order of the predecessive Kautz digraph. For the prefix-preserved shrink strategy, each node $x = x_1x_2...x_{D-1}x_D$ constructs a temporary routing table by the following operations:

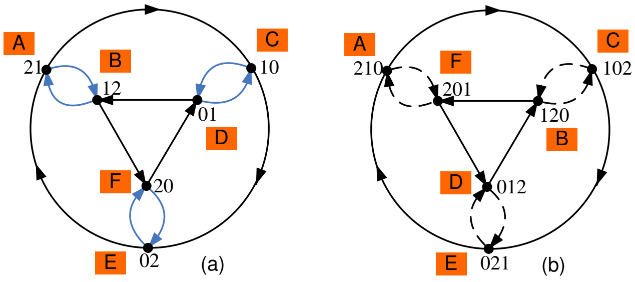


Fig. 4. The topology of MOORE before and after expanding the topology if using the prefix-preserved expansion strategy.

1) Updates its logical identifier with $x_1x_2...x_{D-1}$.
2) Updates the logical identifier $y_1y_2...y_{D-1}y_D$ of its $(i+1)^{th}$ out-neighbor node with $y_1y_2...y_{D-1}$ where $0 \leq i < d$.
3) Discovers the physical address of a node which updates its logical identifier $y_1y_2...y_{D-1}y_D$ with $y_1y_2...y_{D-1}$.

For $0 \leq i < d$, the new and original $(i+1)^{th}$ neighbor of node x are not necessarily the same node. Actually, only one neighbor of node x does not change after performing the topology shrink operation. The node x therefore must discover the physical address of each new neighbor by routing a query to the node. After all existing nodes finish those operations, each of them updates its routing table with the temporary routing table, and finally updates links according to its new routing table. As an example, Figure 4 (b) becomes Figure 4 (a) after performing this type of topology shrink operation. Theorem 5 proves the network overhead of this operation.

Theorem 5: In the case of the prefix-preserved shrink strategy, the shrink of the entire overlay network results in $n \times (d-1)\log_d n$ additional network overhead.

Proof: As discussed above, each node must explore physical addresses of $d-1$ new neighbors by initiating $d-1$ query messages. It is clear that each of these messages will be routed to a destination within at most $\log_d n$ hops. Therefore, the total number of messages caused by expanding the overall topology is at most $n \times (d-1)\log_d n$. Thus, Theorem 5 holds. □

5.3 Suffix-preserved adjustment strategy

As mentioned in Theorems 4 and 5, the topology expansion and shrink operations based on the prefix-preserved strategy suffer from large network overhead. To address this problem, we adopt the suffix-preserved strategy. In this situation, the expansion of the entire topology is implemented by the following local operations at each existing node $x = x_1x_2...x_D$ in MOORE.

1) Updates its logical identifier with $\langle \sigma_1^{-s}(x), x \rangle$.

2) Updates the logical identifier of its $(s+1)^{th}$ out-neighbor node $\sigma_1^s(x)$ with $\langle x, \sigma_1^s(x)\rangle$.
3) Updates the identifier of its $(i+1)^{th}$ out-neighbor node $\sigma_1^i(x)$ with $\langle \sigma_1^{-s}(\sigma_1^i(x)), \sigma_1^i(x)\rangle$, where $0 \le i < i$ and $i{\neq}s$.

After finishing this kind of topology expansion, resources at each node must be transferred to another node if we keep on distributing resources based on the longest-prefix matching policy. To avoid costly movements of resources among nodes during the process of expanding the topology, MOORE distributes resources according to the longest suffix matching policy instead of the longest prefix matching policy.

A resource labeled $x_l...x_D...x_2x_1$ is stored and maintained by its preferred host labeled $x_D...x_2x_1$ if this node exists in MOORE. Otherwise, the resource will be taken over by its second host labeled $\langle \sigma_1^{-s}(x_{D-1}...x_2x_1), x_{D-1}...x_2x_1\rangle$. The topology construction and maintenance strategies ensure that at least the second host of each resource appears in MOORE. In this case, Theorem 6 shows that each resource stays at the original node after expanding the overall topology.

Theorem 6: In the case of the suffix-preserved expansion strategy, the expansion of the entire network does not cause additional network overhead, except d^D+d^{D-1} messages to start the process.

Proof: In the case of MOORE based on a Kautz digraph $K(d,D)$, each resource $x_l...x_D...x_2x_1$ is hosted by its preferred node $x{=}x_D...x_2x_1$. After expanding the overall topology of MOORE, node x updates its identifier with $\langle \sigma_1^{-s}(x), x\rangle = x_{D+1}x_D...x_2x_1$. Node x is still the preferred host of resources whose identifiers have a suffix $x_{D+1}x_D...x_2x_1$, and becomes the second host of other resources stored in it before expanding the topology. Therefore, each resource stays at the original node after expanding the topology, and does not introduce any overhead.

On the other hand, each node $x{=}x_D...x_2x_1$ maintains links to its out-neighbors $x_{D-1}...x_1\alpha$ where $\alpha{\in}\{0,1,2,...,d\}{-}\{x_1\}$. After expanding the topology, the node obtains a new logical identifier $\langle \sigma_1^{-s}(x), x\rangle = x_{D+1}x_D...x_2x_1$, and maintains links to nodes $x'_D...x_2x_1\beta$, where $\beta{\in}\{0,1,2,...,d\} - \{x_1\}$ and the value of x'_D obeys to the topology construction rule of the quasi-Kautz digraph mentioned above. For $\forall\beta{\in}\{0,1,2,...,d\}-\{x_1\}$, the identifer of an out-neighbor $x'_D...x_2x_1\beta$ of node $x_{D+1}x_D...x_2x_1$ is $x_{D-1}...x_2x_1\beta$ before expanding the topology. It is clear that all out-neighbors of node $x_{D+1}...x_2x_1$ are the same out-neighbors of node $x_D...x_2x_1$ although their logical identifiers are updated.

In other words, the links maintained by each node do not change, and no network overhead is further incurred. For example, node A is labeled 21, and has out-neighbor B with identifier 12 and C with identifier 10, before expanding the entire topology, as shown in Figure 5 (a). After expanding the entire topology, the identifiers of node A, B and C are updated as 021, 012, and 210, respectively. As shown in Figure 5 (b), the out-neighbors of node A are still the nodes B and C. Thus, Theorem 6 holds. □

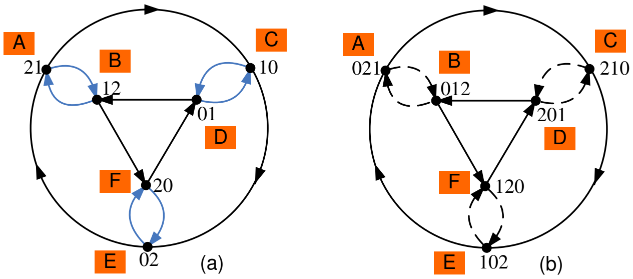


Fig. 5. The topology of MOORE before and after expanding the topology if using the suffix-preserved expansion strategy.

In the case of the suffix-preserved strategy, the shrink of the entire topology is implemented by the following local operations at each existing node $x = x_1x_2...x_D$.

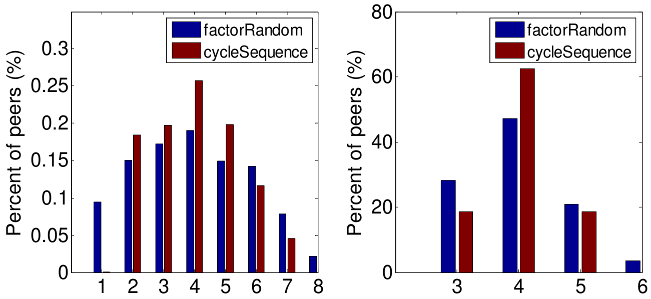
1) Updates its logical identifier with $x_2x_3...x_D$.
2) Updates the logical identifier $y_1y_2...y_D$ of its $(i{+}1)^{th}$ out-neighbor node with $y_2y_3...y_D$ where $0 \le i < d$.
3) Discovers the physical address of a node which updates its logical identifier $y_1y_2...y_D$ with $y_2y_3...y_D$.

After all existing nodes finish those operations, each of them updates its routing table with the temporary routing table, and finally updates links according to its new routing table. The longest suffix matching policy of resource distribution ensures that each resource stays at the original node after shrinking the overall topology.

Theorem 7: In the case of the suffix-preserved shrink strategy, the shrink of the entire overlay network does not cause additional network overhead, except d^D+d^{D-1} messages to start the process.

Proof: Each existing node $x{=}x_D...x_2x_1$ maintains links to its out-neighbors $x'_{D-1}...x_1\alpha$ where $\alpha{\in}\{0,1,2,...,d\}{-}\{x_1\}$ and the value of x'_D obeys Definition 1. After shrinking the topology, the node obtains a new logical identifier $x_{D-1}...x_2x_1$, and maintains links to nodes $x_{D-2}...x_2x_1\beta$, where $\beta{\in}\{0,1,2,...,d\} - \{x_1\}$. For $\forall\alpha{\in}\{0,1,2,...,d\} - \{x_1\}$, the identifier of an out-neighbor $x'_{D-1}...x_2x_1\alpha$ of node $x{=}x_D...x_2x_1$ is updated with $x_{D-2}...x_2x_1\alpha$ after shrinking the topology. It is clear that all out-neighbors of node $x_{D-1}...x_2x_1$ are the same out-neighbors of node $x_D...x_2x_1$ although their logical identifiers are updated.

In other words, the links maintained by each node do not change, and no network overhead is further incurred. For example, node A is labeled 021, and has out-neighbor B with identifier 012 and C with identifier 210 before shrinking the entire topology, as shown in Figure 5 (b). After shrinking the entire topology, the identifiers of node A, B and C are updated as 21, 12, and 10, respectively. As shown in Figure 5 (a), the out-neighbors of node A are still the node B and C. Thus, Theorem 7 holds. □



(a) The in degree of peers in a network with order 7680.

(b) The in degree of peers in a network with order 18000.

Fig. 6. The in-degree distribution of $IK(4, 7680)$ and $IK(4, 18000)$.

6 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. Our simulations are cycle-based, and the MOORE topology with any order 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 routing 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 rounds of simulations.

6.1 Degree distribution of MOORE

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

Proof: The proof has been proposed in Section 3. □

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

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 $K(4, 6)$ and $4n_0$ equals that of $K(4, 7)$. Thus, the least in-degree of $IK(4, 7680)$ and $IK(4, 18000)$ are 1 and 3 according to Theorem 3, as shown in Figure 6. 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.

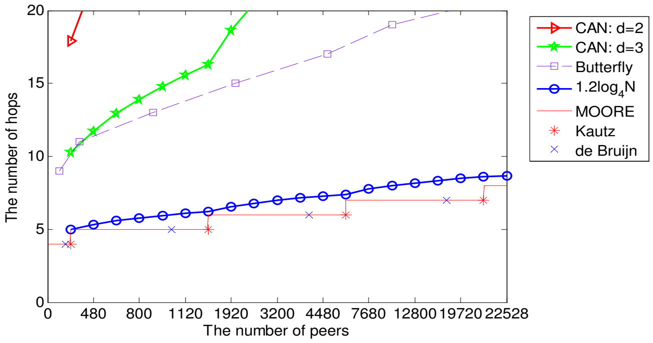


Fig. 7. The diameter of several topologies under different configurations.

6.2 Diameter and path length distribution of MOORE

In an overlay network, the length of a routing path denotes the number of hops from the source to the destination along the routing path.

Property 2: Given a MOORE with arbitrary order n and out-degree d, its diameter is $D_l = \lceil \log_d \frac{n}{d+1} + 1 \rceil$.

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

According to the well known results of the *order/diameter* problem, we know that D_l is the smallest diameter for any number of vertices n where $d^{D-1}+d^{D-2} \leq n \leq d^{D}+d^{D-1}$. A lookup for a resource or node initiated by any node can reach its destination in $O(\log_d n)$ hops. The same result holds for publishing resources.

We evaluate the diameter and average path length of MOORE in different scales (from 256 peers to 22528 peers) and compare it with other constant degree digraphs with the same degree 4, such as 2-dimensional CAN, 3-dimensional CAN, 4-dimensional butterfly, de Bruijn, and Kautz digraph. 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 $n'(n-1)$ routings.

As shown in Figures 7 and 8, the curves of butterfly, de Bruijn and Kautz digraphs are dashed lines or discrete points since their orders are discrete sequences, while that of MOORE and CAN are solid lines because of their arbitrary orders. In Figure 7, the diameter of MOORE is less than $1.2 \log_4 n$, and that of butterfly and CAN at the whole *order* axis. In Figure 8, the average path length of MOORE is also less than $1.2 \log_4 n$, and that of butterfly and CAN at the whole *order* axis. In the two figures, we do not compare MOORE with k-dimensional CCC directly since the degree of CCC is 3 irrespective of the value of k. In reality, the diameter and average

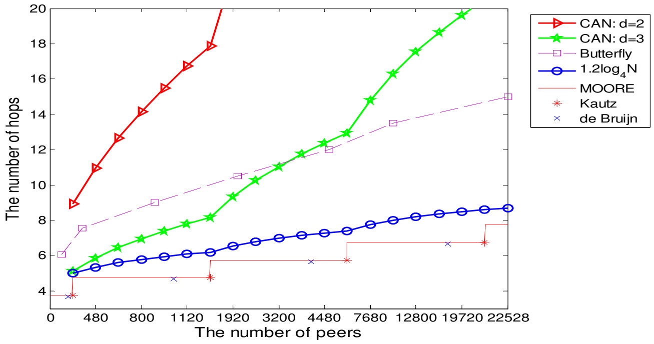


Fig. 8. The average path length of several topologies under different configurations.

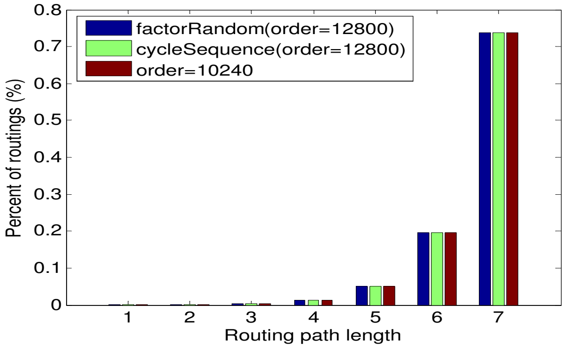


Fig. 9. The path length distribution of $IK(4, 12800)$ and $IK(4, 10240)$.

path length of MOORE with out-degree 3 are also less than that of CCC, respectively. Furthermore, the average path length of MOORE under different scales is trivially different if the scales are covered by an identical range, such as [320,1280), [1280,5120), [5120,20480) in Figure 8.

Property 3: With the shortest path routing scheme, MOORE can achieve low congestion.

Proof: Figure 9 shows the distribution of the routing path length of $IK(4, 12800)$ and $IK(4, 10240)$. We can observe that more than 90% of routing path lengths 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 long path routing scheme used by [10], [19]. Therefore, it is reasonable that MOORE also can achieve the similar low congestion characteristic discussed by Xu et al. [12] and Li et al. [19], although our algorithm adopts a shortest path routing scheme. □

Property 4: Messages caused by node joining and departing operations are at most $2.5d \log_d n$ and $(2.5d + 1) \log_d n$. Only d and $2d$ nodes need to update routing tables when dealing with a new node and a departed node, respectively.

Proof: Recall that Algorithm 3 must find d out-neighbors in order to construct its routing table, and inform d in-neighbors to update their routing table. Algorithm 4 may need to find a substitute node first. Therefore, the former part of Property 4 holds because the routing length is less than $1.2 \log_d n$, and the latter part also holds according to the two algorithms. □

Ideally there should also be a discussion on one of the biggest problems of P2P systems, i.e. performance under churn. This is partially addressed through the discussion in section 5; though, which level of churn would still be sustainable for MOORE is not being discussed.

7 Conclusion

MOORE is the first efficient structured P2P network based on the quasi-Kautz digraph, and is $O(\log_d n)$ in diameter with a constant node out-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 types of queries such as range and multi-attribute queries, and consider the locality of the physical network to reduce latency.

Acknowledgments

The authors would like to thank anonymous reviewers for their constructive comments. The work of Deke Guo is supported in part by the NSF China under grant No. 60903206, and the Military Equipment Preliminary Research Foundation of China under Grant No. 9140A06050610KG0117. The work of Hai Jin is supported in part by the NSFC/RGC Joint Research Foundation under Grant 60731160630. Yunhao Liu's research is supported in part by the NSF CNS-0832120, the NSF China under grants No. 60828003, No. 60773042 and No. 60803126, the National Basic Research Program of China (973 Program) under grant No. 2010CB328100 and No. 2006CB30300, the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z180.

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 degree-optimal distributed hash table. In *Proc. International Peer-to-Peer Symposium*, pages 98–107, Berkeley, CA, USA, February 2003.

[7] J. Xu. *Topological Structure and Analysis of Interconnection Networks*. Kluwer Academic Publishers, 2001.

[8] S. Banerjee and D. Sarkar. Hypercube connected rings: A scalable and fault-tolerant logical topology for optical networks. *Computer Communications*, 24(11):1060–1079, 2001.

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

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

[11] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph. Tapestry: A fault-tolerant wide-area application infrastructure. *ACM SIGCIMM Computer Communication Review*, 32(1):81–81, 2002.

[12] 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.

[13] M. G. Hluchyj and M. J. Karol. Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks. In *Proc. IEEE INFOCOM*, page 379C390, New Orleans, Louisiana, USA, 1988.

[14] H. Shen, C. Xu, and G. Chen. Cycloid: A constant-degree and lookup-efficient P2P overlay network. *Performance Evaluation*, 63(3):195–216, 2006.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

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

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

[23] N. Alon, S. Hoory, and N. Linial. The Moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002.

[24] R. M. Damerell. On Moore graphs. In *Proc. Cambridge Philosophical Society*, pages 227–236, 1973.

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

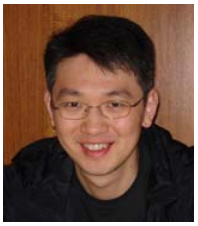
[26] D. Guo, Y. Liu, and X. Li. BAKE: A balanced kautz tree structure for peer-to-peer networks. In *Proc. 27th IEEE INFOCOM*, April 2008.

[27] Y. Zhang, D. Li, and X. Lu. Distributed line graphs: A universal framework for building DHTs based on arbitrary constant-degree graphs. In *Proc. IEEE ICDCS*, pages 152–159, June.

[28] 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.

Deke Guo received the B.S. degree in industry engineering from Beijing University of Aeronautic and Astronautic, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2008. He was a visiting scholar at the Department of Computer Science and Engineering in Hong Kong University of Science and Technology from Jan. 2007 to Jan. 2009. He is now an assistant professor of Information System and Management, National University of Defense Technology, Changsha, China. His current research interests include peer-to-peer computing, Bloom filters, data center networking, and wireless networks. He is a member of the ACM and the IEEE.

Jie Wu is currently a distinguished professor in the Department of Computer Science and Engineering, Florida Atlantic University. He has been on the editorial board of the IEEE Transactions on Parallel and Distributed Systems and was a guest editor of Computer and the Journal of Parallel and Distributed Computing. He is currently on the editorial board of the IEEE Transactions on Mobile Computing. He was a program cochair of the First IEEE International Conference on Mobile Ad Hoc and Sensory Systems (MASS 2004), the executive program vice chair of the 28th IEEE International Conference on Distributed Computing Systems (ICDCS 2008), and the program vice chair of the 29th International Conference on Parallel Processing (ICPP 2000). He was also the general chair of MASS 2006 and is the general chair of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008). He has served as a distinguished visitor of the IEEE Computer Society and is the chairman of the IEEE Technical Committee on Distributed Processing (TCDP). His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 450 papers in various journals and conference proceedings. He is the author of Distributed System Design (CRC Press). He is the recipient of the 1996-1997, 2001-2002, and 2006-2007 Researcher of the Year Awards from Florida Atlantic University. He is a fellow of the IEEE.



Yunhao Liu (SM'06) received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.A. degree from the Beijing Foreign Studies University, China, in 1997, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is now an Associate Professor and the Postgraduate Director at the Department of Computer Science and Engineering in the Hong Kong University of Science and Technology. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. Dr. Liu and his student Mo Li received the Grand Prize of Hong Kong ICT Best Innovation and Research Award 2007. He is a member of the ACM and a senior member of the IEEE.

Hai Jin received the Ph.D. degree in computer engineering in 1994 from Huazhong University of Science and Technology (HUST), China, where he is currently the Cheung Kong Professor and the dean of the School of Computer Science and Technology. In 1996, he was awarded a German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz in Germany. He worked at the University of Hong Kong between 1998 and 2000, and as a visiting scholar at the University of Southern California between 1999 and 2000. He was awarded the Distinguished Young Scholar Award from the National Science Foundation of China in 2001. He is the chief scientist of ChinaGrid, the largest grid computing project in China. He is a senior member of the IEEE and a member of the ACM. He is the member of Grid Forum Steering Group (GFSG). He has coauthored 15 books and published more than 400 research papers. His research interests include computer architecture, virtualization technology, cluster computing and grid computing, peer-to-peer computing, network storage, and network security.

Hanhua Chen is a Ph.D. candidate in the School of Computer Science and Technology at the Huazhong University of Science and Technology (HUST) in China. His research interests include peer-to-peer computing, information retrieval, and wireless sensor network. Hanhua is a student member of the IEEE and the IEEE Computer Society. He was awarded the Microsoft Fellowship in 2005.

Tao Chen received the B.S. degree in military science and the M.S. degree in military operational research from National University of Defense Technology, Changsha, China, in 2004 and 2006, respectively. He is currently working toward the Ph.D. degree in the School of Information System and Management, National University of Defense Technology, Changsha, China. His current research interests include wireless sensor networks, peer-to-peer computing, and data center networking. He is a student member of the IEEE.