

# BAKE: A Balanced Kautz Tree Structure for Peer-to-Peer Networks

Deke Guo<sup>\*†</sup>, Yunhao Liu<sup>†</sup>, and Xiangyang Li<sup>‡</sup>

<sup>\*</sup> College of Information Systems and Management, National University of Defense Technology, Changsha, China

<sup>†</sup> Department of Computer Science and Engineering, Hong Kong University of Science and Technology

<sup>‡</sup> Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616

**Abstract**—In order to improve scalability and reduce maintenance overhead for structured Peer-to-Peer systems, researchers design optimal architectures with constant degree and logarithmic diameter. The expected topologies, however, require the number of peers to be some given values determined by the average degree and the diameter. Hence, existing designs fail to address the issue due to the fact that 1) we cannot guarantee how many peers to join a P2P system at a given time, and 2) a P2P system is typically dynamic with peers frequently coming and leaving. In this work, we propose BAKE scheme based on balanced Kautz tree structure with  $\log_{d+1}$  diameter and constant degree even the number of peers is an arbitrary value. Resources that are similar in single or multi-dimensional attributes space are stored on a same peer or neighboring peers. Through formal analysis and comprehensive simulations, we show that BAKE achieves optimal diameter and good connectivity as the Kautz digraph does. Indeed, the concepts of balanced Kautz tree introduced in this work can also be extended and applied to other interconnection networks after minimal modifications, for example, de Bruijn digraph.

## I. INTRODUCTION

Structured Peer-to-Peer (P2P) models have been proposed as a good candidate infrastructure for building large-scale and robust network applications [1], [2], [3], [4], [5]. They impose a certain structure on the overlay network and control the placement of data, thus exhibit several unique properties that unstructured systems lack. In the design of such networks, the most common concerns are the limitation on peer out-degree and network diameter. The peer out-degree influences the size of routing table to be maintained on each peer. The network diameter indicates the number of hops a lookup needs to travel in the worst case. Traditionally, the peer out-degree and network diameter increase logarithmically with respect to the size  $N$  of a network, such as Chord [2] and Pastry [6]. Those schemes can publish and lookup resources within  $O(\log N)$  hops. They, however, introduce huge maintenance overhead and suffer from poor scalability.

To address this issue, researchers propose architectures based on interconnection networks [1], [3], [7], [8], [9]. In those designs, the network diameter increases logarithmically with respect to the size of the P2P system, while the out-degree of each peer can be a constant, setting a better tradeoff between routing table size and routing delay. The critical requirement of those designs is that the number of peers must be some given values determined by the peer degree and the network diameter. Hence, the approaches are often impractical

in real implementations, especially when considering peers are frequently coming and leaving [10], [11].

In this study, we aim at designing a novel P2P architecture with smaller diameter and constant peer degree even the number of peers is an arbitrary value. Thus, the scheme is practically easy to implement without being restricted by peer dynamics. After looking into literatures, we observe that Kautz digraph is the best choice among existing non-trivial digraphs since it almost achieves Moore Bound [12].

We propose a robust and efficient P2P network, BAKE, based on *balanced Kautz tree* introduced in this paper. BAKE attains the optimal network diameter and constant peer out-degree in a dynamic environments. The main contributions of this paper are as follows:

- 1) We propose a balanced Kautz tree structure, and then design BAKE: an effective and robust P2P architecture which retains desirable properties of static Kautz digraph, such as optimal diameter and constant out-degree.
- 2) We design some algorithms to deal with peer join/departure, and topology changes.
- 3) We evaluate the topology properties of BAKE, the robustness of routing scheme, and the delay and message cost of basic operations through formal analysis and comprehensive simulations.

The rest of this paper is organized as follows. Section II presents *balanced Kautz tree*. Section III discusses the design of BAKE based on balanced Kautz tree. Section IV presents the dynamic operations to maintain the topology. We evaluate BAKE in Section V, and conclude the work in Section VI.

## II. KAUTZ TREE STRUCTURE

**Definition 1:** A  $d$ -ary Kautz tree with depth  $k$  is a rooted tree. The root node has  $d+1$  child nodes, and each inner node has at most  $d$  children. Each edge at same level is assigned a unique label. Each node except the root node is given a unique label. The label of a node is the concatenation of the labels along the edges on its root path. The label of each edge is assigned based on the following rules.

- 1) The edge from the root node to its  $i$ th child is labeled as  $x_1^i = i - 1$  for  $1 \leq i \leq d + 1$ . The  $i$ th child of root node is labeled as  $x_1^i$ , and is arranged from left to right.
- 2) The edge from a node  $x_1$  to its  $i$ th child is labeled as  $x_2^i = (x_1 - i) \bmod (d + 1)$  for  $1 \leq i \leq d$ . The  $i$ th child is labeled as  $x_2^i x_1$ , and is arranged from left to right.

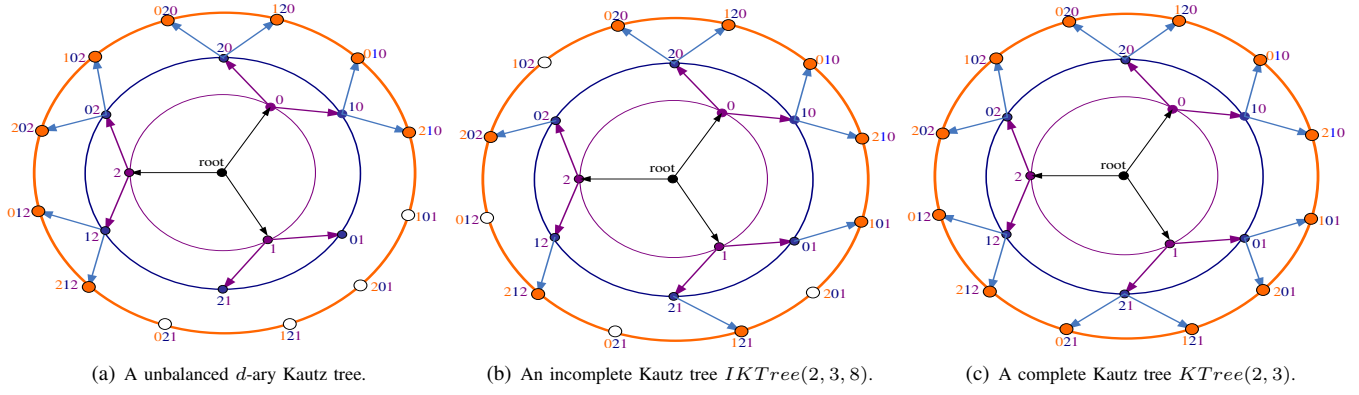


Fig. 1. Three major categories of Kautz tree.

- 3) The edge from a node  $x_{k-1}...x_2x_1$  to its first child is labeled as  $x_k^1$  where  $x_k^1 = x_1$  if  $x_1 \neq x_{k-1}$ , otherwise  $x_k^1 = (x_1 - 1) \bmod (d + 1)$ . The first child is labeled as  $x_k^1 x_{k-1}...x_2x_1$ , and is arranged at the leftmost position.
- 4) We arrange the values  $0, 1, \dots, d$  along a ring in ascend order. If a path from point  $x_k^1$  to point  $x_k^i = (x_k^1 - i + 1) \bmod (d + 1)$  along the anti-clockwise direction does not meet  $x_{k-1}$ , the edge from  $x_{k-1}...x_2x_1$  to its  $i$ th child is labeled as  $x_k^i = (x_k^1 - i + 1) \bmod (d + 1)$ , otherwise it is labeled as  $x_k^i = (x_k^1 - i) \bmod (d + 1)$ . The  $i$ th child of node  $x_{k-1}...x_2x_1$  is labeled as  $x_k^i x_{k-1}...x_2x_1$  for  $2 \leq i \leq d$ , and is arranged from left to right.

The identifier  $x_k x_{k-1}...x_2x_1$  of each node in a Kautz tree satisfies that  $x_{i+1} \neq x_i$  for  $1 \leq i \leq k - 1$ . We associate each node in the tree with a unique level. The level of the root node is 0, its immediate child nodes locate at level 1, and so on. In general, the length of the label of each node represents its level in the tree. Definition 1 provides the mechanism of assigning labels to edges and nodes, however, does not require which child nodes of each inner node appear in the tree. As a result, there are many different shapes of Kautz tree which satisfy definition 1 and have same number of nodes. In this paper, we focus on balanced Kautz tree as shown in Fig.1.

**Definition 2:** A  $d$ -ary Kautz tree with depth  $k$  is balanced if all leaf nodes are at the level  $k$ . A balanced kautz tree is a complete Kautz tree  $KTree(d, k)$  only if the parent node of any leaf node has full child nodes, otherwise it is an incomplete Kautz tree  $IKTree(d, k, n)$  with  $n$  leaf nodes.

In a  $d$ -ary balanced Kautz tree with depth  $k$ , the root node has  $d + 1$  children, each inner node at level  $k-1$  has at least one child and at most  $d$  child nodes, each inner node at other level has  $d$  child nodes, and the number of inner nodes at level  $i$  is  $d^i + d^{i-1}$  for  $1 \leq i \leq k-1$ . The number of leaf nodes in an incomplete Kautz tree is at least  $d^{k-1} + d^{k-2}$  which equals to that of a complete Kautz tree  $KTree(d, k-1)$ , and at most  $d^k + d^{k-1}$  if it becomes a complete Kautz tree  $KTree(d, k)$ . Figure.1(a) plots an unbalanced Kautz tree, and Fig.1(b) and 1(c) plot an incomplete Kautz tree and a complete Kautz tree, respectively.

**Definition 3:** Given a node  $x = x_k...x_2x_1$ ,  $\sigma_1^i(x)$  produces a label for its  $i$ th child node such that  $\sigma_1^i(x) = x_{k+1}^i x_k...x_2x_1$  for

$1 \leq i \leq d$ . If holds one of the following conditions,  $x_{k+1}^i = (x_1 - i + 1) \bmod (d + 1)$ , otherwise  $x_{k+1}^i = (x_1 - i) \bmod (d + 1)$ .

- 1)  $x_k < x_1 - i + 1 \leq x_1$
- 2)  $x_1 < x_k$  and  $x_k - d - 1 < x_1 - i + 1$

$$\sigma_m^1(x) = \sigma_1^1(\sigma_{m-1}^1(x)) \quad (1)$$

$$\sigma_m^d(x) = \sigma_1^d(\sigma_{m-1}^d(x)) \quad (2)$$

The operations  $\sigma_m^1(x)$  and  $\sigma_m^d(x)$  denote the leftmost and rightmost nodes when traversing down  $m$  steps from node  $x$ . The operation  $\sigma_0^i(x)$  denotes the peer  $x$  itself for any  $i$ . The traversal process always selects the first child in each step to arrive at the leftmost node, and chooses the last child in each step to reach the rightmost node. This operation can allocate an unique label for each child node, as well as ranking all the child nodes in an ascend order.

**Definition 4:** For any node  $x$ , its predecessor is the last existing node anti-clockwise from it in a Kautz ring of existing nodes at same level, and its successor is the first existing node clockwise from it in the same Kautz ring. The left adjacent node  $Ladjacent(x)$  is similar to the predecessor, but the Kautz ring is consisted of all possible nodes not just existing nodes. So do the right adjacent  $Radjacent(x)$  and successor node.

### III. BAKE: A BALANCED KAUTZ TREE BASED OVERLAY

First, each peer maps to one leaf node in a balanced Kautz tree, and uses label of its related leaf node and IP address as its logical and physical identifiers, respectively. Second, each peer maintains  $d + 2$  neighbor peers according to a topology rule mentioned below. Third, any resource gets an identifier from an identifier space. Resources are distributed at given peers based on the longest suffix matching rule. Based on the above three strategies, we propose a robust routing scheme to support different operations effectively.

#### A. Topology construction rule

For a balanced Kautz tree  $KTree(d, k)$ , number of  $d^k + d^{k-1}$  peers form a topology by the following rule. For each peer  $x = x_k...x_2x_1$ , its successor and predecessor are peers  $Radjacent(x)$  and  $Ladjacent(x)$ , and its  $i$ th out-neighbor and in-neighbor are  $\varsigma_1^i(x)$  and  $\tau_1^i(x)$  for  $1 \leq i \leq d$ , respectively. The  $\varsigma_1^i(x)$  and  $\tau_1^i(x)$  are defined as follows.

*Definition 5:*  $\varsigma_1^i(x)$  denotes an operation such that  $\varsigma_1^i(x) = x_{k-1} \dots x_1 x_0^i$  for  $1 \leq i \leq d$ . If one of the following conditions is satisfied, then  $x_0^i = (x_k + i - 1) \bmod (d + 1)$ . Otherwise,  $x_0^i = (x_k + i) \bmod (d + 1)$ .

- 1)  $x_k < x_k + i - 1 \leq x_1$ ;
- 2)  $x_k + i - 1 < x_1 + d + 1$  and  $x_1 < x_k$ .

*Definition 6:* Given  $\sigma_1^i(x) = x_{k+1}^i \dots x_2 x_1$ ,  $\tau_1^i(x)$  denotes an operation such that  $\tau_1^i(x) = x_{k+1}^i \dots x_3 x_2$  for  $1 \leq i \leq d$ .

For a balanced Kautz tree  $IKTree(d, k, n)$ ,  $n$  peers form an overlay network based on the following rules. Each peer  $x = x_k \dots x_2 x_1$  maintains links to its predecessor, successor and  $d$  out-neighbors. The predecessor and successor are identified during runtime. The  $d$  out-neighbors are the peers satisfying one of the following conditions in order. For  $1 \leq i \leq d$ :

- 1) If a peer  $\varsigma_1^i(x)$  has appeared in the overlay network, it is the  $i$ th out-neighbor of peer  $x$ ;
- 2) Otherwise, if  $\varsigma_1^i(x)$  and its predecessor  $y$  have a common suffix with length  $k - 1$ , peer  $y$  is the  $i$ th out-neighbor.
- 3) Otherwise, if  $\varsigma_1^i(x)$  and its successor  $z$  have a common suffix with length  $k - 1$ , peer  $z$  is the  $i$ th out-neighbor.

*Theorem 1:* The above construction rules can guarantee that any peer  $x = x_k \dots x_2 x_1$  in BAKE based on  $IKTree(d, k, n)$  has  $d$  out-neighbors besides its successor and predecessor.

Later in Section IV, we show that the topology management operations make sure that  $i$ th neighbor of each peer  $x$  is available in a static environment. In reality, it might be difficult because of peer failures in highly dynamic scenarios. To deal with the negative impacts of failed peers, a flexible peer joining strategy is introduced to make the unavailable out-neighbor of peer  $x$  becomes available as soon as possible.

#### B. Resource placement based on longest suffix matching

For any resource to be distributed in BAKE, it gets a long identifier  $x = x_1 \dots x_k \dots x_2 x_1$  based on its value of single or multiple attributes. If a peer  $x_k \dots x_2 x_1$  exists, it is the preferred host of resource  $x$ , otherwise a selected peer with a suffix  $x_{k-1} \dots x_2 x_1$  acts as the second host. In a static environment, the preferred or second host peer must exist. In a dynamic environment, the peers with  $x_{k-1} \dots x_2 x_1$  as suffix may fail concurrently such that we can not select any node. To address this issue, we define the predecessor peer of  $x_k \dots x_2 x_1$  as the third host of resource  $x$ .

#### C. Robust and effective routing

Fiol et al. propose a shortest path routing scheme for a similar routing problem [13], and we also introduce an improved and practical routing scheme in previous works [14]. Generally, a peer  $x$  finds its largest suffix  $u$  that appears as a prefix of  $y$ , then forwards message to a neighbor  $z$  such that its largest suffix  $v$  coincides with a prefix of  $y$  and the length of  $v$  is larger than that of  $u$ . Such schemes work well in a static environment, but suffer from poor robustness in dynamic environment. For example, peer 202 routes a message to peer 101 along the expected path  $202 \rightarrow 121 \rightarrow 210 \rightarrow 101$  in Fig.1(b), and fails if one peer in the path becomes unavailable.

In BAKE, we allow peers to send a message towards another neighbor when fails to route it along the expected path.

When a peer  $x = x_k x_{k-1} \dots x_2 x_1$  publishes or lookups a resource with identifier  $y_1 \dots y_k \dots y_2 y_1$ , the preferred destination is peer  $y = y_k \dots y_2 y_1$ . If such a peer  $y$  does not exist, the message will be forwarded to its second/third host. For example, in Fig.1, peer 010 routes a resource with identifier 012021 towards peer 021 along a path  $010 \rightarrow 202 \rightarrow 021$ , and peer 202 will find that the preferred host peer 021 does not exist. It then forwards the message to second host 121. If peer 202 finds the peer 121 also fails, it routes the message towards peer 201 along a path  $202 \rightarrow 020 \rightarrow 201$ . The resource is finally stored by peer 101 identified by peer 020. The resource placement strategy guarantees any resource is stored by an existing third host if the first and second preferred hosts are absent. Note that the decision on new destination of a message can be made based only on local knowledge at each peer.

### IV. TOPOLOGY MANAGEMENT

#### A. Topology adjustment

A native BAKE based on an initial  $KTree(d, k)$  can be constructed in advance. When more peers want to join, we expand  $KTree(d, k)$  to achieve a  $IKTree(d, k + 1, d^k + d^{k-1})$ . If number of peers in BAKE reaches  $(d + 1) \times (d^k + d^{k-1})$ , the incomplete Kautz tree becomes a complete Kautz tree and is ready to be expand further.

To expand the topology of BAKE, we introduce an efficient solution that needs only local operations. For an existing peer with identifier  $x = x_k x_{k-1} \dots x_2 x_1$  in BAKE, we

- 1) Update its logical identifier with  $\sigma_1^1(x)$ .
- 2) Update the logical identifier of its successor peer  $x' = x'_k x_{k-1} \dots x_2 x_1$  with  $\sigma_1^1(x')$ . The logical identifier of its predecessor peer is updated similarly.
- 3) Update the logical identifier of its out-neighbor peer  $\varsigma_1^i(x)$  with  $\sigma_1^1(\varsigma_1^i(x))$  for  $1 \leq i \leq d$ .

*Theorem 2:* The topology expanding process of the overlay network does not cause additional network overhead except  $d^k + d^{k-1}$  messages to start the process.

In contrast to expand the overlay, BAKE shrinks the topology by the following local operations when the number of existing peers decreases to the number of leaf nodes in a  $d$ -ary complete Kautz tree. For a existing peer  $x = x_{k+1} \dots x_2 x_1$ , we replace its logical identifier with  $x_k \dots x_2 x_1$ , and update the logical identifiers of its predecessor, successor, and  $d$  out-neighbors in a same way.

#### B. Peer joins

The peer related to the leftmost leaf node  $\sigma_{k-1}^1(0)$  in a  $d$ -ary incomplete Kautz tree with depth  $k$  acts as the first entry point of BAKE, and its predecessor acts as a synchronous second entry point. The entry points first serve as general peers, and also manage all the labels of leaf nodes in the incomplete Kautz tree. The entry point allocates a leaf node to a peer as follows. First, if a peer fails and recovers in time, the entry point allocates the previous leaf node to the peer if that leaf is not assigned yet. Second, it allocates the leaf node  $x$  which

is the first child node of its parent node, then the leaf node which is the second child node of its parent node, and so on. Third, for the leaf nodes with same position value, it allocates them in the clockwise order at the Kautz ring of their parent nodes. If all leaf nodes are allocated, the first entry point starts the topology expanding process. It may also start the topology shrink process when required.

A new peer should consult the first entry point for its logical identifier  $x = x_k \dots x_2 x_1$ , predecessor  $y$  and successor  $z$ , and construct its topology by the following process. The peers having  $x_{k-1} \dots x_2 x_1$  as a common suffix possess the same out-neighbors. If there exists at least one of such peers, peer  $x$  can get a copy of the out-neighbors from one of them. Otherwise, peer  $x$  discovers its out-neighbors based on the topology rules in the worst case. Indeed, all in-neighbors of each peer  $\zeta_1^i(x)$  for  $1 \leq i \leq d$  are not available, and there is no routing path along in-neighbor links to these peers. Thus peer  $x$  routes a query message towards the left or right adjacent peer of  $\zeta_1^i(x)$  for  $1 \leq i \leq d$  along predecessor or successor links. Peer  $x$  also informs peer  $\tau_1^i(x)$  for  $1 \leq i \leq d$  to update an out-of-date neighbor with it. The message is routed towards a peer  $u = \tau_1^1(x)$ , and forwarded to other related peers along the successor links.

### C. Peer departures

For a peer  $x = x_k \dots x_2 x_1$  that is about to leave, if its predecessor  $y$  or successor  $z$  has a suffix  $x_{k-1} \dots x_2 x_1$ , it notifies peer  $y$ , peer  $z$  and the first entry point of BAKE. Peer  $y$  will update its successor with  $z$ . Peer  $z$  will replace its predecessor with peer  $y$ . If peer  $y$  has a suffix  $x_{k-1} \dots x_2 x_1$ , peer  $x$  transfers its resources to peer  $y$  before it departs, and notifies its in-neighbor peer  $\tau_1^i(x)$  to replace the link to it with another link to peer  $y$  for  $1 \leq i \leq d$ . If peer  $y$  does not have a suffix  $x_{k-1} \dots x_2 x_1$  while peer  $z$  has, peer  $x$  transfers its resources to peer  $z$ , and notifies its in-neighbor peer  $\tau_1^i(x)$  to replace the link to it with another link to peer  $z$  for  $1 \leq i \leq d$ .

If none of peer  $y$  and peer  $z$  has a suffix  $x_{k-1} \dots x_2 x_1$ , peer  $x$  should find a substituted peer  $w$  before it leaves. It detects other peers with a suffix  $x_{k-1} \dots x_2 x_1$ , and selects the peer whose position value is the largest one if there are at least one such peers, otherwise it consults the first entry point of BAKE for peer  $w$ . Peer  $w$  must satisfy two constraints: 1) There are other peers that have a common suffix, length  $k-1$ , with peer  $w$  in BAKE; 2) Its rank among all peers having the common suffix should be the largest. Peer  $w$  will perform a voluntary departure operation, and takes over the resources, logical identifier, and routing table of peer  $x$ . It then informs neighbors of peer  $x$  to update physical identifier of peer  $x$ .

## V. ANALYSIS AND EVALUATION

We first emulate the evolution process of BAKE from an initial overlay network based on  $KTree(d, 1)$  through those dynamic operations of peers and other topology management operations in PeerSim [15]. We then analyze and evaluate the topology properties, the robustness of routing scheme, and the delay and message cost of major operations.

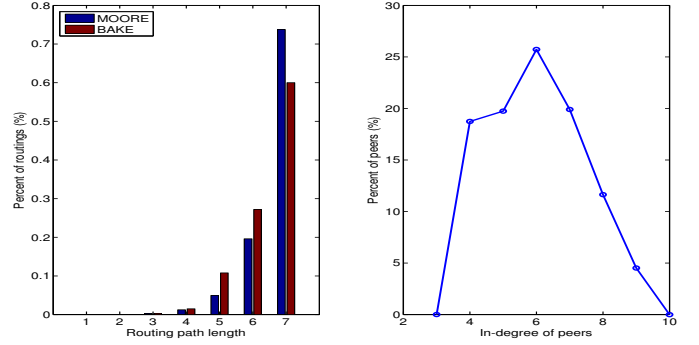


Fig. 2. The routing and in-degree distributions of IKTree(4,7,12800).

### A. Topology properties

**Theorem 3:** The out-degree of each peer is  $d + 2$  in BAKE with  $n$  peers, and the diameter is  $k_l = \lceil \log_d(n) - \log_d(1 + 1/d) \rceil$ .

The diameter does not reflect the entire view of routing scheme, hence, we study the expected value and distribution of the length of routing paths among many pairs of nodes. In simulations, each peer first issues a message towards others. As shown in the left sub-figure of Fig.2, the length of about 60% routing paths equals to the diameter. If we compare BAKE with MOORE [14] under the same configurations, the length of less routing paths equals to the diameter, and that of more routing paths is less than the diameter. The right sub-figure of Fig.2 indicates that the in-degree of most peers is adjacent to  $d + 2$ , and that of other peers is less than  $d + 2$ .

The expected value of routing path is denoted as  $ard$ . We evaluate the  $ard$  of BAKE where  $n$  ranges from 320 to 23040 and  $d$  equals to 4. We compare BAKE with MOORE and other constant degree topologies in which the out-degree of each peer is four, such as CAN, 4-dimensional butterfly, de Bruijn, and Kautz digraph. Fig.3 plots the experiment results. The curves of butterfly, de Bruijn and Kautz digraphs are dashed lines and discrete points since their orders are special discrete sequences. The curves of BAKE and MOORE are solid lines since their orders can be arbitrary values. The  $ard$  of BAKE and MOORE are shorter than that of butterfly, CAN,  $\log_4 n$  and de Bruijn, and the simulation results also confirm this. The  $ard$  of BAKE is less than that of MOORE, and is more close to that of Kautz digraph. The  $ard$  of Kautz digraph with shortest path routing scheme has been proved in [16]. The  $ard$  of BAKE is a little less than that of Kautz digraph when  $n$  equals  $d^i + d^{i-1}$  for  $1 \leq i$ , because partial routing paths in Kautz digraph are shortened with help of the successor and predecessor links of each peer. Hence, BAKE achieves an optimal topology which inherits all good properties of Kautz digraph even its order is out of  $d^k + d^{k-1}$  for a given  $d$  and any  $k > 1$ .

### B. Robustness of routing

The routing schemes mentioned in [14], [13] suffer from poor robustness in dynamic environments. BAKE addresses this issue by sending a message to another out-neighbor, if it fails to forward the message along a shortest path. The connectivity of each peer in BAKE is  $d + 2$ , implying that,

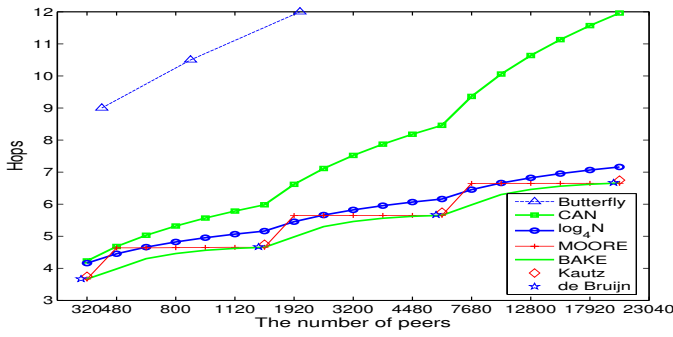


Fig. 3. The average routing distance under different configurations.

as long as  $d + 2$  neighbors and/or links do not fail simultaneously, a message can be delivered to another available peer successfully. In words, when  $d$  is a modest value, any message can reach its destination with high probability even if peers fail randomly, as shown in Fig.4. BAKE makes sure that a large majority of messages can be routed successfully when number of  $d$  even  $2d$  peers fail randomly, and outperforms existing schemes.

### C. Delay and message costs of basic operations

We define  $\alpha = \lceil n / (d^{k_l-1} + d^{k_l-2}) \rceil$  for later analysis. For a message to lookup or publish a resource with identifier  $x = x_k \dots x_2 x_1$ , the routing delay is at most  $t(k_l - 1) + k_l$  hops before it reaches an available host  $y = y_k \dots y_2 y_1$ , where  $t$  denotes the anti-clockwise distance from node  $x_{k-1} \dots x_2 x_1$  to  $y_{k-1} \dots y_2 y_1$  in the given Kautz ring. In a static system, delay of all queries is  $k_l$  because peer  $y$  is always the preferred host. In a moderately dynamic system, delay of a majority of queries is  $k_l$  because peer  $y$  usually is the preferred or second host. In a highly dynamic system, delay of lots of queries is  $t(k_l - 1) + k_l$  because peer  $y$  is often a third host peer, where  $0 \leq t$  but  $t$  is not a large value.

**Theorem 4:** In a static or moderately dynamic environment, the delay and message cost of a new peer  $x$  joining is at most  $2k_l + \alpha + 1$ .

**Corollary 1:** In a highly dynamic environment, the delay of a new peer  $x$  joining is at most  $3k_l + \alpha + 2$  hops. The whole process causes at most  $3 \times (k_l + \alpha)$  messages.

**Theorem 5:** The delay and message cost to handle a leaving peer is at most  $2k_l + \alpha + 2$ .

## VI. CONCLUSION

We propose a balanced Kautz tree structure, based on which we design BAKE, a practical P2P architecture. The topology management and routing schemes guarantee flexible and efficient resource distribution. BAKE achieves optimal diameter, high performance, and good connectivity for dynamic P2P networks.

## VII. ACKNOWLEDGMENTS

The work of Deke Guo was supported in part by Ph.D. Candidates Research Foundation of National University of Defense Technology under Grant No.0615. E-mail: guodk@cse.ust.hk. The work of Yunhao Liu was supported

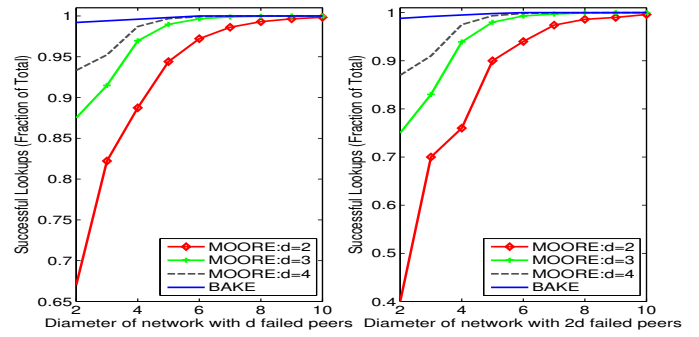


Fig. 4. The fraction of successful lookups as a function of the fraction of peers that fail concurrently in MOORE under different scales.

in part by Hong Kong RGC Grant N HKUST614/07 and HKUST 615206/E, NSF China under grants No.60573140 and 60673179, National High Technology R&D Program of China (863) under grant No.2007AA01Z180, and Nokia APAC research grant. E-mail: liu@cs.ust.hk.

## 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] 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.
- [4] 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.
- [5] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph. Tapestry: a fault-tolerant wide-area application infrastructure. *Computer Communication Review*, 32(1):81, 2002.
- [6] 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.
- [7] H. Shen, C. Xu, and G. Chen. Cycloid: A constant-degree and lookup-efficient p2p overlay network. In *Proc. the 18th International Parallel and Distributed Processing Symposium*, Santa Fe, New Mexico, USA, April 2004.
- [8] P. Fraigniaud and P. Gauron. D2B: a de bruijn based content-addressable network. *Theor. Comput. Sci.*, 355(1):65–79, 2006.
- [9] 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.
- [10] M. Ripeanu, A. Iamnitchi, and I. T. Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1):50–57, 2002.
- [11] Yunhao Liu, Li Xiao, and Lionel M. Ni. Building a scalable bipartite p2p overlay network. *IEEE Trans. Parallel Distrib. Syst.*, 18(September):1296–1306, 2007.
- [12] 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.
- [13] 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.
- [14] D. Guo, J. Wu, H. Chen, and X. Luo. Moore: An extendable peer-to-peer network based on incomplete kautz digraph with constant degree. In *Proc. 26th IEEE INFOCOM*, page 821, May 2007.
- [15] 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.
- [16] G. Panchapakesan and A. Sengupta. On a lightwave networks topology using kautz digraphs. *IEEE Computer*, 48(10):1131–1138, 1999.