Information Processing Letters 110 (2010) 821–825

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

KCube: A novel architecture for interconnection networks

Deke Guo [a], Hanhua Chen [b], Yuan He [c], Hai Jin [b], Chao Chen [a], Honghui Chen [a,*], Zhen Shu [a], Guangqi Huang [a]

[a] *Key Laboratory of Science and Technology for C⁴ISR Technology, School of Information System and Management, National University of Defense Technology, Changsha Hunan 410073, China*

[b] *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan Hubei 430074, China*

[c] *Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China*

ARTICLE INFO

Article history:
Received 1 July 2009
Received in revised form 27 May 2010
Accepted 27 June 2010
Communicated by A. Tarlecki

Keywords:
Interconnection networks
Compound graph
Kautz digraph
Hypercube graph
Hierarchical networks

ABSTRACT

This paper proposes a novel architecture called KCube. KCube is a compound graph of Kautz digraph and hypercube. It employs the hypercube topology as a unit cluster and connects many such clusters by means of a Kautz digraph. It then utilizes the topological properties of hypercube to realize convenient embedding of parallel algorithms, and the short diameter of Kautz graph to support efficient inter-cluster communication. KCube possesses many attractive characteristics, such as modularity, expansibility, and regularity, while these benefits are achieved at the cost of only increasing the degree of any node by one, regardless of the network size. The methodology to construct KCube can also be applied to other compound networks after minimal modifications.

1. Introduction

Interconnection networks become popular in digital systems, offering effective and economical solutions to optimize the communication and interconnection among system components. In the literature of interconnection networks (see [1]), hypercube has been proved to be one of the most popular architectures, due to its attractive properties such as strong connectivity, regularity, topological symmetry, and recursive constructions. The degree of each node in hypercube, however, increases logarithmically with respect to the network size. Such a fact makes the direct use of hypercube prohibitive in large-scale network applications, e.g. data centers [2–4].

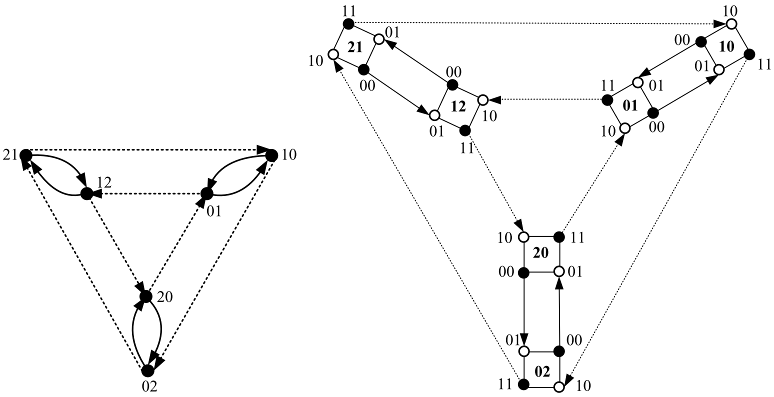
The hierarchical network is a natural way to construct large networks, where many small basic networks in the lower level(s) are interconnected at higher level(s). Many schemes have been proposed to perform various graph operations on those small networks. The main schemes include overlay, join, product, composition, compound, and complete bipartite graphs [5]. Among such operations, the compound graphs are observed to be suitable for large-scale digital systems, which possess good regularity and expansibility. For example, a compound graph requires only one additional link per node at level 1 to support communication among the level-2 nodes. This helps to reduce the cost of expansion when one intends to increase the network size.

In this paper we propose a novel architecture of two-level interconnection networks, called KCube, and the methodology to construct it. KCube is a compound graph of Kautz digraph and hypercube. It incarnates the good characteristics of the Kautz digraph and hypercube. Specifically, it utilizes the topological properties of hypercube to realize convenient embedding of parallel algorithms, and utilizes the shortest diameter of Kautz digraph to support efficient inter-cluster communications.

Among the state-of-arts schemes, dBCube [6] is closest to our work in this paper. dBCube is a compound graph of de Bruijn digraph and hypercube, which is obtained by re-

* Corresponding author. Tel.: +86 731 84576603.
E-mail address: chh0808@gmail.com (H. Chen).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.06.010



(a) $K(2,2)$ (b) $KC(2,2)$

Fig. 1. $K(2,2)$ and $KC(2,2)$.

placing each node in de Bruijn digraph with a hypercube cluster. Compared to dBCube, KCube provides a general construction methodology, which can be applied to other compound networks, for example dBCube which lacks the construction methodology in [6]. In addition, KCube generates a network with smaller diameter under the same node degree and network size. Moreover, KCube generates a larger network than dBCube does, under the same node degree and network diameter.

2. KCube network

In this section, we present the methodology to synthesizing a KCube network. Multiple I/O ports are present at each node to connect with the other nodes through links. A link connecting two nodes in the same cluster is called a local link, while a link connecting two nodes in difference clusters is called a remote link. The connection topology inside each cluster is a hypercube, while the interconnection topology at the level of clusters is a Kautz digraph.

Section 2.1 summarizes the notations and definitions used in this paper. Section 2.2 presents the methodology to compose a KCube network. A routing algorithm tailored to the KCube network is described in Section 2.3. Section 2.4 illuminates the topological properties of KCube.

2.1. Notation and definitions

Let the interconnection network be modeled by an graph $G(V,E)$, where the set of vertices V, represents the processors in the network and the set of edges E, represents the communication links in the network. In the rest of this paper, we use the terms network and graph, node and vertex, link and edge, interchangeably.

The m-dimension *hypercube* graph is denoted by $H(m)$, where $m \geqslant 1$. The vertex set of $H(m)$ is $\{x_m \ldots x_i \ldots x_1\}$, where $x_m \ldots x_i \ldots x_1$ denotes a sequence and $x_i = 0$ or 1 for all $1 \leqslant i \leqslant m$. There is an edge between any two vertices if and only if their labels differ by exactly one bit. There are 2^m vertices in $H(m)$. The node degree and network diameter are m.

The *de Bruijn* digraph [7], $D(d,k)$, has node out-degree of d and network diameter of k, where $d \geqslant 1$ and $k \geqslant 1$. The vertex set is $\{x_k \ldots x_i \ldots x_1 \mid x_i \in \{0,1,\ldots,d-1\}$ for all $1 \leqslant i \leqslant k\}$, where $x_k \ldots x_i \ldots x_1$ denotes a sequence. There is an arc from vertex $x_k x_{k-1} \ldots x_1$ to vertex $x_{k-1} \ldots x_1 \alpha$ for each $\alpha \in \{0,1,\ldots,d-1\}$.

The *Kautz* digraph [8,9], $K(d,k)$, has node out-degree d and network diameter k, where $d \geqslant 1$ and $k \geqslant 1$. The vertex set is $\{x_k \ldots x_i \ldots x_1 | x_i \in \{0,1,\ldots,d\}$ and $x_i \neq x_{i+1}$ for all $1 \leqslant i \leqslant k\}$, where $x_k \ldots x_i \ldots x_1$ denotes a sequence. There is an arc from vertex $x_k x_{k-1} \ldots x_1$ to vertex $x_{k-1} \ldots x_1 \alpha$ for each $\alpha \in \{0,1,\ldots,d\} - x_1$.

For any node in $D(d,k)$ or $K(d,k)$, the in-degree and out-degree are the same d. There are d^k and $d^k + d^{k-1}$ vertices in $D(d,k)$ and $K(d,k)$, respectively. It is clear that $K(d,k)$ possesses more vertices than $D(d,k)$, especially for large d and/or k. In addition, there are $d(d^k + d^{k-1})$ arcs in $K(d,k)$. Fig. 1(a) is an example of a Kautz digraph, $K(2,2)$.

2.2. Construction methodology of KCube

Definition 1. Given two regular graphs G_2 and G_1, the compound graph $G_2(G_1)$ is obtained by replacing each node of G_2 by a copy of G_1 and replacing each link of G_2 by a link which connects corresponding two copies of G_1.

KCube is the compound graph of a Kautz digraph G_2 and a hypercube G_1, where the Kautz digraph and hypercube graph have been proved to be regular. It employs the hypercube topology as a unit cluster and connects many such clusters by means of Kautz digraph. In the resultant graph, the topology of G_2 is preserved and only one link is inserted to connect two copies of G_1. There is an additional remote link associated with each node in a hypercube cluster. For each node in the resultant network, the degree is identical. A constraint must be satisfied for the two basic graphs to constitute a compound graph. The degree of G_2 must be equal to the number of nodes in G_1.

A KCube is characterized by $KC(m,d,k)$, where m is the dimension of the hypercube graph, d is the node out-degree, and k is the diameter of the Kautz digraph. According to the aforementioned constraint of KCube, we can infer that the degree of each node in the Kautz digraph should be 2^m. We know that the out-degree and in-degree of each node in $K(d,k)$ are d. Thus, the degree of each

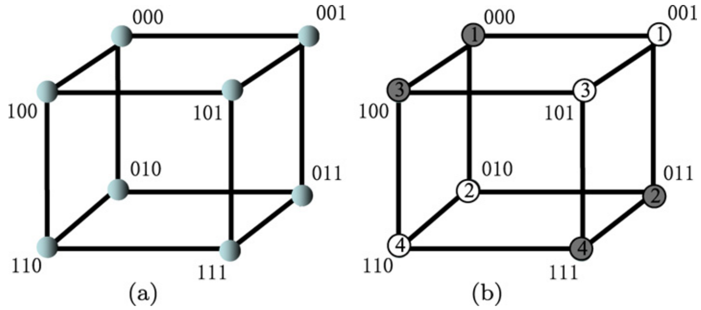


Fig. 2. The partition and sort of all nodes in $H(3)$.

node in $K(d,k)$ is $2d$, and $d = 2^{m-1}$. For simplicity, a KCube is characterized by $KC(m,k)$ in the remainder of this paper.

In $KC(2,2)$, each block represents a cluster consisting of $H(2)$ and the degree of the Kautz digraph connected those hypercubes equals to the number of nodes in $H(2)$, as shown in Fig. 1(b). In $KC(2,2)$, the number of $H(2)$ clusters is $2^{k(m-1)} + 2^{(k-1)(m-1)} = 6$ and the number of total nodes is $2^{km} + 2^{k(m-1)+1} = 24$. Note that only one remote arc is associated with each node in $H(2)$.

Recall that KCube should replace each link of $K(2^{m-1}, k)$ by a link connecting two nodes that belong to different copy of $H(m)$. To realize this goal, the following two preconditions must be satisfied in advance.

1. The out-degree and in-degree of each node in $K(2^{m-1}, k)$ are 2^{m-1} and each hypercube cluster contains 2^m nodes. Thus, 2^{m-1} nodes in $H(m)$ need to be selected as the heads of remote arcs which are incident to corresponding nodes in the other clusters. Such nodes are referred to as *output nodes* of $H(m)$. The remaining 2^{m-1} nodes act as the tails of the remote arcs which are incident from corresponding nodes in the other clusters. Such nodes are referred to *input nodes* of $H(m)$.
2. There are $d(d^k + d^{k-1})$ remote arcs in $KC(m,k)$. These arcs need to be mapped to pairs of nodes, where one end of an arc is an *output node* of a cluster and the other end is an *input node* of another cluster.

To satisfy the first precondition, we need an approach to divide all nodes of $H(m)$ into two equal parts. We use an arbitrary node $x_m \dots x_2 x_1$ from $H(m)$ as a reference node. The set of *output nodes* of $H(m)$ consists of all nodes $y_m \dots y_i \dots y_2 y_1$, where

- $y_2 y_1 = x_2 x_1$ or $y_2 y_1 = \bar{x}_2 \bar{x}_1$.
- $y_i = 0$ or 1 for all $3 \leqslant i \leqslant m$.

The set of *input nodes* of $H(m)$ consists of all nodes $y_m \dots y_2 y_1$, where

- $y_2 y_1 = \bar{x}_2 x_1$ or $y_2 y_1 = x_2 \bar{x}_1$.
- $y_i = 0$ or 1 for all $3 \leqslant i \leqslant m$.

The only property of the cluster-splitting approach used is that complementary nodes are of the same type. An approach which randomly selects pairs of complementary nodes as input nodes would also allow Lemma 1 to hold. For other random approaches which divides all nodes into two equal parts, Lemma 1 does not hold. For example, if nodes 000, 100, 110, and 011 in Fig. 2 are randomly selected as the input nodes, other nodes are the output nodes, the largest length of the shortest path between an input node and an output node becomes m, not $m-1$ proved in Lemma 1. Our approach further results in a shorter path between any two nodes in a KCube digraph than the random approach.

For ease of presentation, we choose node $x_m \dots x_2 x_1$ as the reference node where $x_i = 0$ for all $1 \leqslant i \leqslant m$. Thus, the nodes whose last two bits of label are 00 or 11 constitute the set of output nodes of $H(m)$, and the nodes whose last two bits of label are 01 or 10 constitute the set of input nodes of $H(m)$. All nodes in $H(3)$ as shown in Fig. 2(a) are partitioned into two equal sets. Those black balls denote the output nodes, while those white balls denote the input nodes, as shown in Fig. 2(b). To satisfy the second precondition, we further sort all the output nodes of $H(m)$ in the ascending order of the node labels and all the input nodes of $H(m)$ in the same way. For example, the output nodes of $H(m)$ are sorted in the order of 000, 011, 100, 111, and sort all the input nodes of $H(m)$ are sorted in the order of 001, 010, 101, 110, as shown in Fig. 2(b).

We then sort all the out-arcs and in-arcs of any node in $K(2^{m-1}, k)$ in the ascending order with the following approach. We can infer from the definition of Kautz digraph that for a node $x_k \dots x_2 x_1$, its out-arc to node $x_{k-1} \dots x_1 \alpha$ for $\alpha \in \{0, 1, \dots, d\} - \{x_1\}$ is denoted as the ith out-arc. Here, i indicates the clockwise distance from x_k to α (if $x_1 \neq x_k$) or from $x_k + 1$ to α (if $x_1 = x_k$) in a ring consisting of the values $0, 1, \dots, d$ in the ascending order. It is worth noticing that the ith out-arc of any node $x_k \dots x_2 x_1$ is also the ith in-arc of a corresponding node $x_{k-1} \dots x_1 \alpha$. Thus, all 2^{m-1} out-arcs of each node can be sorted in the ascending order, and all 2^{m-1} in-arcs of each node can be sorted in the same way.

Subsequently, we can constitute a compound graph $KC(m,k)$ on the basis of multiple hypercubes $H(m)$ and $K(2^{m-1}, k)$ through the following approach. First, any given node x in $K(2^{m-1}, k)$ is replaced by a hypercube $H(m)$. Second, the ith out-arc of the node x is replaced by a remote arc between the ith output node of a hypercube and the ith input node of another hypercube. Here, the two hypercubes correspond to the head and tail of the ith out-arc of the node x in $K(2^{m-1}, k)$. Consequently, any node in

Algorithm 1 Routing algorithm in $KC(m,k)$

1: According to the simple self-routing of Kautz digraph [8] and the labels of two hypercube clusters x and x', node $\langle x,y\rangle$ determine the next hypercube cluster u the message should be forwarded to. In the Kautz digraph which connects all hypercube clusters in $KC(m,c)$, we can derive the order i of the out-arc from node x to node u among all out-arcs of node x.
2: According to the construction rule of $KC(m,k)$, the message should be forward from the ith output node in hypercube x to the ith input node in the hypercube u. If node $\langle x,y\rangle$ is not the ith output node in the hypercube x, it infers the label of the ith output node in the same hypercube, and routes the message to the ith output node using the self-routing of hypercube [10]. Otherwise, it forwards the message directly to the first input node of the hypercube u.

$KC(m,k)$ is labeled $\langle x = x_k \dots x_2x_1,\ y = y_m \dots y_2y_1\rangle$, where $x_k \dots x_2x_1$ is called the Kautz-part-label and $y_m \dots y_2y_1$ is called the hypercube-part-label. Given any node, it has a remote arc to only one node within different hypercube clusters referred to as Kautz-part-neighbor. Meanwhile, it has local arcs to m nodes in the same cluster referred to as hypercube-part-neighbors. Fig. 1 shows a representation of the resultant $KC(2,2)$. The black balls and white balls represent the output nodes and input nodes in each hypercube cluster, respectively. The solid and dot directed edges are the first and second out-arcs of each hypercube cluster. As shown in Fig. 1, the ith out-arc of each hypercube cluster interconnects the ith output node and ith input node of two corresponding hypercube clusters. An undirected edge interconnects two nodes in the same hypercube cluster.

2.3. Routing in KCube networks

For efficiency of communication, a simple and fast routing algorithm should ensure that a message can be forwarded from a source node $\langle x,y\rangle$ to a destination node $\langle x',y'\rangle$ along a shortest path. In *KCube*, routing can be implemented by Algorithm 1, which systematically integrates the existing routing algorithms for hypercube and Kautz graphs. For example, a message from node $\langle 21,11\rangle$ to node $\langle 20,01\rangle$ is first routed to node $\langle 21,00\rangle$ which is the first output node in hypercube 21. It is clear that there are two candidate paths between the two nodes, i.e. $\langle 21,11\rangle \to \langle 21,10\rangle \to \langle 21,00\rangle$ or $\langle 21,11\rangle \to \langle 21,01\rangle \to \langle 21,00\rangle$. The message is then routed to node $\langle 20,10\rangle$ along a path $\langle 21,00\rangle \to \langle 12,01\rangle \to \langle 12,11\rangle \to \langle 20,10\rangle$. Finally, the message is routed to the destination $\langle 20,01\rangle$ along one of the two possible paths $\langle 20,10\rangle \to \langle 20,11\rangle \to \langle 20,01\rangle$ or $\langle 20,10\rangle \to \langle 20,00\rangle \to \langle 20,01\rangle$.

In reality, Algorithm 1 cannot derive the shortest pathes for all pairs of nodes. For example, the shortest path from $\langle 21,11\rangle$ to $\langle 20,01\rangle$ is $\langle 21,11\rangle\to\langle 10,10\rangle\to\langle 10,11\rangle\to\langle 02,10\rangle\to\langle 02,00\rangle\to\langle 20,01\rangle$. As a result, Algorithm 1 only ensures that the upper bound on the diameter of Fig. 1(b) is 7 while the diameter is 6.

2.4. Topology properties

Theorem 1. *In a $KC(m,k)$, there is $2^{k(m-1)} + 2^{(k-1)(m-1)}$ hypercube clusters and $2^{k(m-1)+m} + 2^{k(m-1)+1}$ vertices.*

Proof. From the definition of $KC(m,d,k)$, we know that the number of hypercube clusters is $d^k + d^{k-1}$ and the number of vertices in each hypercube cluster is 2^m. Theorem 1 holds due to the fact that $d = 2^{m-1}$. □

Lemma 1. *The largest length of the shortest path between an output node and an input node in the same hypercube $H(m)$ is $m-1$.*

Proof. In $H(m)$, the largest number of nodes which must be traversed in order to travel from any node $x_m \dots x_2x_1$ to node $\bar{x}_m \dots \bar{x}_2\bar{x}_1$ is m. The length of the shortest path between the node $x_m \dots x_2x_1$ and every node except the node $\bar{x}_m \dots \bar{x}_2\bar{x}_1$ is less than m. According to the aforementioned approaches, we can infer that the node $x_m \dots x_2x_1$ is included in the set of output nodes or the set of input nodes, together with the node $\bar{x}_m \dots \bar{x}_2\bar{x}_1$. Thus, the largest length of the shortest path between an output node and an input node in the same hypercube $H(m)$ is $m-1$. □

Theorem 2. *An upper bound on the diameter of $KC(m,k)$ is $2m + (k-1)(m-1) + k = m(k+1) + 1$.*

Proof. In $KC(m,k)$, the shortest path from an arbitrary node $\langle x,y\rangle$ to any node $\langle x',y'\rangle$ traverses at most $k+1$ hypercube clusters, including the source hypercube x, destination hypercube x', and other $k-1$ intermediate hypercubes. This is guaranteed by the diameter of $K(2^{m-1},k)$.

In the source hypercube x, the largest length of a shortest path from the node $\langle x,y\rangle$ to any other node is less than or equal to m, and is only equal to m when the node $\langle x,y\rangle$ is an output node and the other node is $\langle x,\bar{y}\rangle$. For any intermediate hypercube along the shortest path from $\langle x,y\rangle$ to $\langle x',y'\rangle$, it receives a message from one of its input node and forward the message to one of its output node within $m-1$ hops, as proved in Lemma 1. For the destination hypercube x', it receives a message from one of its input node and forwards to the node $\langle x',y'\rangle$. The largest length of a shortest path from that input node to the node $\langle x',y'\rangle$ is less than or equal to m, and is only equal to m when that input node is $\langle x',\bar{y}'\rangle$ and the node $\langle x',y'\rangle$ is also an input node of $H_{k+1}(m)$. In addition, the shortest path also traverse k remote links each connects a pair of hypercube clusters. Thus, a upper bound on the diameter of $KC(m,k)$ is $2m + (k-1)(m-1) + k = m(k+1) + 1$, hence Theorem 2 holds. □

Table 1 summarizes various parameters of hypercube, Kautz, and KCube. KCube constructs large networks by connecting multiple copies of smaller hypercube networks at the cost of increasing the degree of each node by only one.

3. Conclusion

This paper presents KCube, a new compound graph of Kautz digraph and hypercube. KCube employs the hypercube topology as a unit cluster and connects many such clusters by means of a Kautz digraph. KCube combines the advantages of hypercube and Kautz graph, and possesses good modularity, expansibility, and regularity. The methodology to construct KCube can also be applied to other compound networks after minimal modifications.

Table 1
Parameters of the networks of interest.

Graph	Network size	Number of edges	Degree	Diameter
$H(m)$	$N_1 = 2^m$	$L_1 = m2^{(m-1)}$	m	m
$K(2^{m-1}, k)$	$N_2 = 2^{(m-1)k} + 2^{(m-1)(k-1)}$	$L_2 = 2^{(m-1)(k+1)} + 2^{(m-1)k}$	2^m	k
$KC(m, k)$	$N_1 \times N_2$	$L1 \times N_2 + L_2 = (m+1)L_2$	$m+1$	$m(k+1)+1$

KCube is a promising architecture for future interconnection networks, such as Peer-to-Peer networks [11–15] and data centers [3,4]. In recent years, many large data centers have been built to provide online application and infrastructure services. A fundamental challenge in data centers is how to efficiently interconnect an increasing number of servers so that one does not need to rewire the existing running servers when adding new servers. KCube is well modular so that it can be enlarged through small increments. Such a capability is dependent upon the number of nodes required at the level of hypercube clusters. It is the ideal case that one can increase the cluster size by one node each time. Although outperforming the existing schemes like dBCube, KCube has a limitation on the hypercube cluster which cannot be realized with an arbitrary size. Actually, to enlarge a hypercube cluster, its size should be doubled. To enlarge a data center, the number of new servers should be equal to the number of existing servers. The new servers are interconnected in the same way as those existing servers. Besides, the new servers and existing servers in the same hypercube cluster are interconnected according to the constructing rule of hypercube. We plan to resolve the limitation in our future work.

Acknowledgements

This work is supported in part by the NSF China under Grants Nos. 60903206, 60803152, the Military Equipment Preliminary Research Foundation of China under Grant No. 9140A06050610KG0117, and the National Basic Research Program of China under Grant No. A0420070009.

References

[1] M.D. Grammatikakis, D.F. Hsu, M. Kraetzl, Parallel System Interconnections and Communications, CRC Press, 2001.

[2] M.A. Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, in: Proc. SIGCOMM, Seattle, Washington, USA, 2008.

[3] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, S. Lu, Dcell: A scalable and fault-tolerant network structure for data centers, in: Proc. SIGCOMM, Seattle, Washington, USA, 2008.

[4] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, S. Lu, Bcube: A high performance, server-centric network architecture for modular data centers, in: Proc. SIGCOMM, Barcelona, Spain, 2009.

[5] D.P. Agrawal, C. Chen, J.R. Burke, Hybrid graph-based networks for multiprocessing, Telecommunication System 10 (1998) 107–134.

[6] C. Chen, D.P. Agrawal, J.R. Burke, dbcube: A new class of hierarchical multiprocessor interconnection networks with area efficient layout, IEEE Transactions on Parallel Distributed Systems 4 (12) (1993) 1332–1344.

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

[8] G. Panchapakesan, A. Sengupta, On a lightwave networks topology using Kautz digraphs, IEEE Transactions on Computers 48 (10) (1999) 1131–1138.

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

[10] Y. Saad, M.H. Schultz, Topological properties of hypercubes, IEEE Transactions on Computers 37 (1988) 867–872.

[11] Y. Liu, L. Xiao, X. Liu, L.M. Ni, X. Zhang, Location awareness in unstructured peer-to-peer systems, IEEE Transactions on Parallel and Distributed Systems 16 (2) (Feb. 2005) 163–174.

[12] C. Wang, L. Xiao, Y. Liu, P. Zheng, Dicas: an efficient distributed caching mechanism for p2p systems, IEEE Transactions on Parallel and Distributed Systems 17 (10) (Oct. 2006) 1097–1109.

[13] Y. Liu, L. Xiao, L.M. Ni, Building a scalable bipartite p2p overlay network, IEEE Transactions on Parallel and Distributed Systems 18 (9) (Sep. 2007) 1296–1306.

[14] Y. Liu, A two-hop solution to solving topology mismatch, IEEE Transactions on Parallel and Distributed Systems 19 (11) (Nov. 2008) 1591–1600.

[15] Y. He, Y. Liu, Vovo: Vcr-oriented video-on-demand in large-scale peer-to-peer networks, IEEE Transactions on Parallel and Distributed Systems 20 (4) (Apr. 2009) 528–539.