Hou R, Wu JG, Chen Y *et al.* Constructing edge-colored graph for heterogeneous networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(5): 1154–1160 Sept. 2015. DOI 10.1007/s11390-015-1551-0

Constructing Edge-Colored Graph for Heterogeneous Networks

Rui Hou[1] (侯 睿), *Student Member, IEEE*, Ji-Gang Wu[1,2,*] (武继刚), *Member, CCF, IEEE*
Yawen Chen[3] (陈亚文), Haibo Zhang[3] (张海波), and Xiu-Feng Sui[2] (隋秀峰), *Member, CCF, IEEE*

[1] *School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China*
[2] *State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences Beijing 100190, China*
[3] *Department of Computer Science, University of Otago, Dunedin 9054, New Zealand*

E-mail: {asrhou, asjgwu}@gmail.com; {yawen, haibo}@cs.otago.ac.nz; suixiufeng@ict.ac.cn

Received May 24, 2014; revised November 8, 2014.

Abstract In order to build a fault-tolerant network, heterogeneous facilities are arranged in the network to prevent homogeneous faults from causing serious damage. This paper uses edge-colored graph to investigate the features of a network topology which is survivable after a set of homogeneous devices malfunction. We propose an approach to designing such networks under arbitrary parameters. We also show that the proposed approach can be used to optimize inter-router connections in network-on-chip to reduce the additional consumption of energy and time delay.

Keywords network reliability, homogeneous fault, fault tolerance, reconfigurable system, network-on-chip

1 Introduction

Network devices (e.g., modems, routers) on the same platform are homogeneous. Communication links with homogeneous network devices can corrupt after a platform dependent computer virus attack, as all devices on the same platform could be infected and stop working in a short time. For example, when a network is attacked by a virus that is developed for a vulnerability of Microsoft Windows operating system, all devices with Windows operating system in the network will break down in a short time. This kind of network faults has been named as homogeneous fault[1]. One of the most challenging problems is designing a fault-tolerant network that can keep the terminals connected after suffering homogeneous faults. Thus, for each terminal, network reliability is implemented by using equipments on heterogeneous platforms to connect different neighbours, instead of adding redundant homogeneous communication links to same neighbours.

Wang and Desmedt[1] modeled such heterogeneous networks as edge-colored graphs in which only the edges that connect vertices with the same technologies (e.g., same software and/or same hardware) have the same color. When all edges with the same color are removed in the edge-colored graph, the corresponding devices are broken down due to the same reason. The authors investigated that if there exist edge-colored graphs under certain constraints, then these edge-colored graphs must have some features in common. They also proved the hardness of the determination of whether a given edge-colored graph is connected within certain thresholds. Wang[2-3] proposed some edge-colored graph designs, in which the number of vertices is not arbitrary, and the constructions in them are based on the perfect factorization of complete graphs. In addition, Wang[2] proved that, given any number of colors, there always exist edge-colored graphs that can keep connected even though any one color is removed. Besides, Paterson *et*

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61173032 and 61202062, the National Natural Science Foundation of China (Tianyuan Special Foundation) under Grant Nos. 11326211 and 11326198, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20131201110002, and the Key Laboratory of Computer Architecture Opening Topic Fund Subsidization under Grant No. CARCH201303.

*Corresponding Author

*al.*① presented some designs on another kind of edge-colored graphs, and provided some conditions that are satisfied. However, very little attention has been paid to a special kind of edge-colored graphs, which are still connected after excluding the edges with one arbitrary color. This motivates us to study how to design such kind of edge-colored graphs.

In [4], "temperature" is utilized to describe the feature of neighbors a vertex has. The correlation between the heat heterogeneity of network structure and the outcome of evolutionary dynamics is investigated as well. But in this paper, we do not focus on the correlation although it is important in complex networks. Our target is on the communication networks that do not change structures frequently, as re-wiring or changing addresses is impractical.

The contributions of this paper are as follows.

1) We prove that edge-colored graphs can be considered as some certain structures that are combined properly if the graphs are still connected after excluding the edges with one arbitrary color.

2) We propose an efficient approach to construct arbitrary shapes of edge-colored graphs for the given number of colors and vertices, and the constructed graphs are still connected after removing edges with any one color.

To the best of our knowledge, this paper is the first work towards constructing a fault-tolerant network which is robust against one homogeneous fault.

The rest of the paper is organized as follows. Section 2 presents the background and definitions used in this paper. Section 3 analyzes the features of the edge-colored graph mentioned above, followed by Section 4, in which the proposed building algorithm is described. We show that the proposed approach can be applied to power and time saving in NoC (network-on-chip) in Section 5. In Section 6, we arrive at some conclusions.

2 Preliminaries

The edge-colored graph is first introduced in [1] to model heterogeneous network, where adjacent edges can have same color.

Definition 1 (Edge-Colored Graph[1]). *An edge-colored graph is a 4-tuple $G(V, E, C, f)$, where V is the set of vertices of different types, E is the set of edges, C is the set of colors, and f is a mapping from E onto C in such a way that the edges that connect the same type of vertices have the same color.*

It is always the case that a communication link in the network can malfunction for a variety of reasons, and such a link can be represented as multiple edges with different colors. Nevertheless, each link in [1] is assumed as a single edge with only one color, and fault-tolerance is realized by appending redundant adjacency that each vertex can communicate with, rather than adding redundant communication links between two vertices which already are neighbors. In this paper, we make the same assumptions.

Definition 2 ($(t+1)$-Color Connectivity[1]). *Two distinct vertices v_a and v_b are $(t+1)$-color connected if there is a path between v_a and v_b in G such that the edges in this path do not have any color in any $C_t \subseteq C$ of size t. An edge-colored graph G is $(t+1)$-color connected if and only if any two vertices in G are $(t+1)$-color connected.*

In other words, if an edge-colored graph is $(t+1)$-color connected, the graph is connected when the edges with t colors are removed. The following problem was studied in [1]: given n vertices, γ colors, and an integer $t \geqslant 1$, how to construct a $(t+1)$-color connected graph using minimum number of edges? The necessary and sufficient conditions are derived if there is a $(t+1)$-color connected graph for four special cases: 1) $\gamma = t+1$; 2) $t = 1$; 3) $\gamma = 4$ and $t = 2$; and 4) $\gamma = 5$ and $t = 3$. The necessary condition for general cases is also derived. In addition, the problem of determining whether a given graph is $(t+1)$-color connected is proven to be co-NP-complete, that is, the complement of an NP-complete problem[5]. A general approach for constructing $(t+1)$-color connected graph was proposed in [3], but this approach only works for the case when the corresponding complete graph K_n has a perfect one-factorization. The work in [2] shows an approach for constructing $(t+1)$-color connected graph when $\gamma = 4$ and $t = 2$. Moreover, the author proved that only 2-color connected graphs exist for $\gamma \geqslant 5$.

An edge-colored graph considered in this paper has at most one edge between two vertices, and it keeps connected when removing at least one color in the graph. It is obvious that an edge-colored graph, which has no more than two vertices, cannot be $(t+1)$-color connected. For an edge-colored graph with three vertices, there are at most three edges in the graph. Moreover, such a graph needs two or more edges to keep con-

①Paterson M, Stinson D, Wang Y. On encoding symbol degrees of array BP-XOR codes. http://coitweb.uncc.edu/~yonwang/papers/bpxorIT.pdf, Feb. 2015.

nected. Hence, a $(t+1)$-color connected graph has more than two colors. Therefore, we assume that $n > 2$ and $\gamma > 2$ in this paper.

A spanning tree is *colorful* if no two of its edges have the same color[6]. We generalize this concept as the following definition for all edge-colored graphs (see Figs.1(a) and 1(b)).

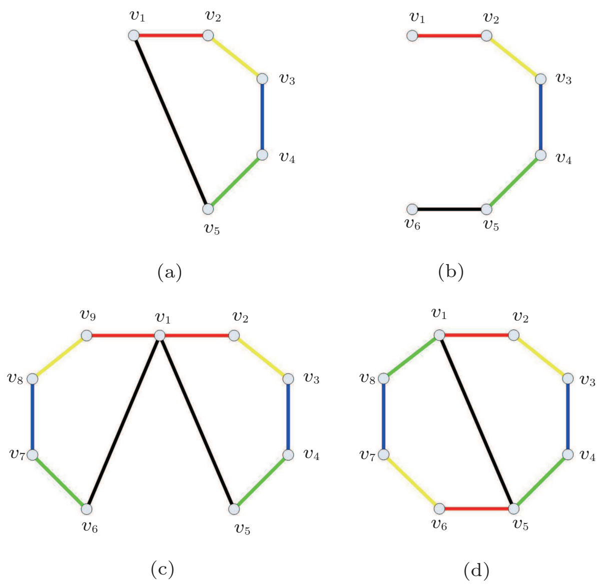


Fig.1. (a) A colorful ring. (b) A colorful path. (c) A 2-color connected graph for $n = 9, m = 10$ and $\gamma = 5$. (d) A 2-color connected graph for $n = 8, m = 9$ and $\gamma = 5$.

Definition 3. *An edge-colored graph is colorful if no two of its edges have the same color.*

3 Analysis of 2-Color Connected Graph

In this section, we analyze a feature of 2-color connected graphs and propose the simplest form of such graphs. Next, we introduce a necessary and sufficient characterization that every 2-color connected graph has. Then, we prove that any 2-color connected graph can be considered as a proper combination of a colorful ring and some colorful paths.

Let E_c indicate the edge set of any one color. If $|E_c| = n-1$, there are enough edges with same color to construct a tree that includes all of the vertices. Then, a 2-color connected graph can be considered as a set of edge-disjoint spanning trees that share common vertices. Thus, the graph keeps 2-color connected because edges with every color can connect all vertices independently, not because edges with any $\gamma - 1$ colors can connect all vertices. In this case, too many edges are required to keep 2-color connected in the graph. As a result, the cost of the corresponding network structure is expensive, and the types of network equipments are limited. Hence, $|E_c| \geqslant n - 1$ is impractical for the network design. We conclude that $|E_c| < n - 1$.

Lemma 1. *Any edge e in a 2-color connected graph belongs to a ring subgraph in which no edge is of the same color as that of e.*

Proof. It is not difficult to understand that a ring is still connected and it becomes a path, after any one edge is removed (see Fig.1(a)). If an edge of an edge-colored graph is not in a ring, then it is in a path of the graph. As shown in Fig.1(b), the graph excluding such an edge is not connected any more. On the other hand, removing any two edges of a ring, the ring will be disconnected, and it becomes two paths or a path and a vertex. Hence, if an edge e in a ring subgraph R of an edge-colored graph G and if another edge in R is of the same color as that of e, then the ring R is not connected after the removing of the color of e, and furthermore, the edge-colored graph G is not 2-color connected. □

It is easy to know that a colorful ring is the simplest 2-color connected graph, from Lemma 1.

Lemma 2. *A colorful ring is a 2-color connected graph.*

Proof. For each edge of a colorful ring, from Definition 3, its color is sole. Therefore, according to Lemma 1, a ring with unique color on each edge is a $(t+1)$-color connected graph, where $t = 1$. □

Lemma 3. *An edge-colored graph is a 2-color connected graph if it consists of a colorful path and a 2-color connected graph, and the colorful path and the graph share the endpoints of the path.*

Proof. Assume a colorful path merges its endpoints v_1 and v_n $(n \neq 1)$ with two different vertices in a 2-color connected graph $G(V, E)$.

In a colorful path, removing a color implies removing an edge base on Definition 3. Thus, if we remove an edge, the path turns out to be two paths (or a vertex and a path in extreme cases) attached to $G(V, E)$ on v_1 and v_n. No matter whether $G(V, E)$ contains edges in the same color or not, the resulting graph $G(V, E')$ is still connected, where $E' \subseteq E$. Hence, the graph including the two paths (or a vertex and a path) and $G(V, E')$ is connected.

If we remove edges of any one color in $G(V, E)$, the new graph $G(V, E'')$ is still connected, where $E'' \subset E$. No matter whether the colorful ring contains edges in the same color or not, it is not difficult to understand that the resulting graph is also connected.

Assume a colorful path merges its endpoints v_1 and v_n $(n \neq 1)$ with one vertex in a 2-color connected graph $G(V,E)$, and the formed graph is a union of $G(V,E)$ and a colorful ring that joins one vertex of $G(V,E)$. Moreover, from Definitions 2 and 3, together with Lemma 2, both $G(V,E)$ and the colorful ring keep connected whenever removing any one color. Hence, the resulting graph is 2-color connected.

From the above discussion, we conclude the lemma. □

The path mentioned in Lemma 3 can be called an "*ear*". Lemmas 2 and 3 support us to construct a 2-color connected graph beginning with a colorful ring, and keep sticking a colorful ear to the ring on one or two vertices (as shown in Figs.1(c) and 1(d)). In fact, arbitrary 2-color connected graph is made up of a colorful ring and some colorful ears.

Lemma 4. *Let $G(V,E)$ be a 2-color connected graph and $H(V',E')$ be the maximal 2-color connected subgraph of $G(V,E)$, where $V' \subset V$ or $E' \subset E$, then $G(V,E)$ is the union of $H(V',E')$ and a colorful path P that merges its endpoints with one or two vertices of $H(V',E')$.*

Proof. It could be that $V = V'$. In this case, $E' \subset E$. Let e be the edge in E but not in E'. The ends of e are in V'. Then P is a monochromatic and colorful path consisting of e. Since $H(V',E')$ is already 2-color connected, no matter what the color of e is, adding P to $H(V',E')$ will not change the connectivity of $H(V',E')$. All vertices are connected by edges in E' after deleting arbitrary color. Therefore, the union of $H(V',E')$ and P is a 2-color connected graph bigger than $H(V',E')$. Taking into account that $H(V',E')$ is the maximal 2-color connected subgraph of $G(V,E)$, the union of $H(V',E')$ and P is $G(V,E)$.

If $V' \subset V$, there is an edge e that one of its ends $v_x \in V'$ and the other $v_y \notin V'$. Considering that $G(V,E)$ is 2-color connected, the graph is connected when removing all edges with the color of e, and there exists a path joining v_y to a vertex in V' without passing e, as e is removed. Suppose v_z (v_z and v_x may be the same vertex) is the first vertex of V' in the path, then a path P, made up of e and the path between v_y and v_z, connects v_x to v_z, and none of vertices in P belongs to V' except v_x and v_z. If P is colorful, from Definition 3, P is broken into at most two parts after deleting an arbitrary color, i.e., all vertices in P are connected to $H(V',E')$ when excluding any one color. Furthermore, $H(V',E')$ is 2-color connected. Hence the union of P and $H(V',E')$ is a 2-color connected graph. Meanwhile, the size of the union is larger than the size of $H(V',E')$. Noting that $H(V',E')$ is the largest 2-color connected subgraph in $G(V,E)$, the union is equal to $G(V,E)$. □

From the above discussion, we have the following theorem.

Theorem 1. *A graph is 2-color connected if and only if it admits a colorful path decomposition to a colorful ring.*

4 Designs for 2-Color Connected Graph

According to our previous analysis, given a 2-color connected graph G, there is a sequence of 2-color connected subgraph

$$H_0 \subseteq H_1 \subseteq H_2 \subseteq \cdots \subseteq H_k = G,$$

where H_0 is a colorful ring and each H_{i+1} is an union of H_i and a colorful ear. In this section, we present the minimum number of edges in a 2-color connected graph on given parameters, and show how to design such a graph.

Theorem 2. *In a 2-color connected graph G, with n vertices and γ colors, the minimum number of edges m in G is*

$$m = \begin{cases} n, & if\ n \leqslant \gamma, \\ \gamma + \gamma\lfloor \frac{n-\gamma}{\gamma-1} \rfloor + \\ \quad (n-\gamma) \bmod (\gamma-1) + 1, & if\ n > \gamma. \end{cases}$$

Proof. As we known, in a graph, the sum of the degrees of the vertices is equal to the twice of the number of edges. For that reason, the smaller m is, the smaller the sum of the degrees of the graph becomes. If $n \leqslant \gamma$, G can be a colorful ring with n vertices or a union of a colorful ring with vertices less than n and some colorful ears. Because the sum of degrees of the union is larger than that of the colorful ring, both of them contain n vertices, and the colorful ring has less edges. Therefore, when $n \leqslant \gamma$, G with minimum m is a colorful ring where $m = n$. If $n > \gamma$, G can only be a union of colorful ring with some colorful ears. In this case, joining less ears to the ring, the sum of the degrees of G is smaller, in other words, the bigger the ring and the ears are, the smaller the sum of the degrees of n vertices is. Taking into account that the biggest colorful ring contains γ vertices and the biggest colorful ear has $\gamma+1$ vertices, if $n = \gamma + i(\gamma-1)$ where $i \in N^*$, G with minimum edges is the union of a colorful ring with γ vertices and i colorful ears, each of which includes $\gamma+1$ vertices, i.e., $m = \gamma + i\gamma$. If $\gamma + i(\gamma-1) < n < \gamma + (i+1)(\gamma-1)$, G with minimum m consists of a colorful ring with γ vertices and $i+1$ colorful ears, accordingly, $m = \gamma + i\gamma + (n-\gamma) \bmod (\gamma-1) + 1$. □

In Algorithm 1 of constructing a 2-color connected graph, we only consider the case of $n > \gamma$. Firstly, we construct a colorful ring, the basic 2-color connected graph, on γ vertices. According to Theorem 1, we establish a colorful path on $\gamma + 1$ vertices and merge the endpoints of the path with one or two vertices of the ring to form a new 2-color connected graph. This procedure is repeated until all vertices are used up, or the colorful path with γ edges cannot be constructed. If some vertices are left, we have no choice but to use all of the rest vertices and the two extra vertices to set up a colorful path. Then we attach it to the 2-color connected graph we built before in the same way.

Algorithm 1. Constructing 2-Color Connected Graph

Input: number of vertices n and colors γ, $n \geqslant \gamma$
Output: a 2-color connected graph with minimum edges

Construct a colorful ring G_1 on γ vertices;
$n = n - \gamma$;
$k = n \mod (\gamma - 1)$;
$i = 1$;
repeat
  $i = i + 1$;
  Set up a colorful path P_i on $\gamma + 1$ vertices;
  Merge the endpoints of P with one or two vertices of G_{i-1} to form a 2-color connected graph G_i;
until $i > k$;
if $n = n - k(\gamma - 1) > 0$ **then**
  Set up a colorful path P on the $n + 2$ vertices;
  Merge the endpoints of P with one or two vertices of G_i to form a 2-color connected graph G_{i+1};
end if

Wang[2-3] proposed 2-color connected graph designs based on the perfect factorization of complete graphs. Hence, the number of vertices in the constructed 2-color connected graph must be certain even, p+1 or 2p, where p is a prime number . The algorithm proposed in this paper is used for building up a 2-color connected graph only with necessary edges. However, from Theorem 1, any 2-color connected graph can be constructed by combining a colorful ring and a set of colorful paths properly, and even a 2-color connected graph on same parameters can be constructed in different shapes.

It is not difficult to understand that the proposed approach can also reduce the edges with some certain colors. In some server-centric data centers, network designs should be scalable, which will spread gradually without a need for re-wiring or changing addresses[7]. Meanwhile, some types of equipment may be expensive or cause high energy consumptions. Thus, the large-scale deployment of such devices in the network is not feasible, due to budget constraints. Consequently, the number of corresponding edges is limited. For this case, we only need to avoid using these edges in our approach when constructing colorful ring and colorful paths, although this may bring an increase in the number of overall edges.

5 Application on Reconfigurable NoC Architecture

In this section, we propose another example application of constructing 2-color connected graphs, and the approach can also be used for tailoring reconfigurable NoC (ReNoC)[8] architecture that minimizes the power and the latency.

Network-on-Chip (NoC) is a general purpose on-chip network that interconnects processing elements (PEs) with routers. As technology advances, an increasing number of PEs are integrated in a single chip, and routers in NoC have to be more complex to support the growth of communication flows. As a result, routers are consuming far more energy than PEs. In ReNoC, the actual physical layout is 2D mesh, and the physical topology can be reconfigured to adjust the application traffic pattern through altering connections between routers. For example, a well-known mapping algorithm NMAP presented in [9] physically maps task graph nodes of input applications on a mesh-based network shown in Fig.2(a), and selects a route for each task graph edge such that the distances between the communicating PEs are minimized. In consequence, Figs.2(b)$\sim$2(e) are application-specific topologies of four input applications set up by NMAP. It follows that inter-router connections in logical topologies are less than those in original physical topology, which greatly reduces the complexity of topology, and thereby reduces the workload of routers. In comparison with static mesh, the evaluation design in [7] shows the power consumption is decreased by 56%. A reconfigurable photonic network on chip (RePNoC) architecture with optical circuit switching is presented in [10], and the transmission speed can be increased by up to nearly doubled, according to the latency performance simulation.

Due to that NoC will run multiple applications and the traffic pattern of each one is various, application-specific topologies of different applications, generated by the same mapping algorithm, are heterogeneous.

To switch between distinctive applications, ReNoC or RePNoC has to change the inter-router connections to adapt to the new one, which is unbearably slow, and thus increases overall run time and additional power consumption. Accordingly, it is necessary to optimize the logical topology of every application to minimize the budget of changing the inter-router connections. As illustrated in Fig.2, each different application's logical topology has some unused links. Suppose that every logical topology makes use of all processing elements in NoC (the mapping algorithm assigns at least one task graph node to a PE), and the unused link sets of the topologies are disjoint, i.e., the topologies are not homogenous. Fig.2 shows that it is possible to design a 2-color connected graph that combines heterogeneous logical topologies of NoC applications, and then the corresponding ReNoC or RePNoC can run many applications continuously on specific topologies, without closing each inter-router connection more than once. For that reason, the entire time of applications running on NoC is shorten if the logical topologies of them can coexist on a 2-color connected graph.

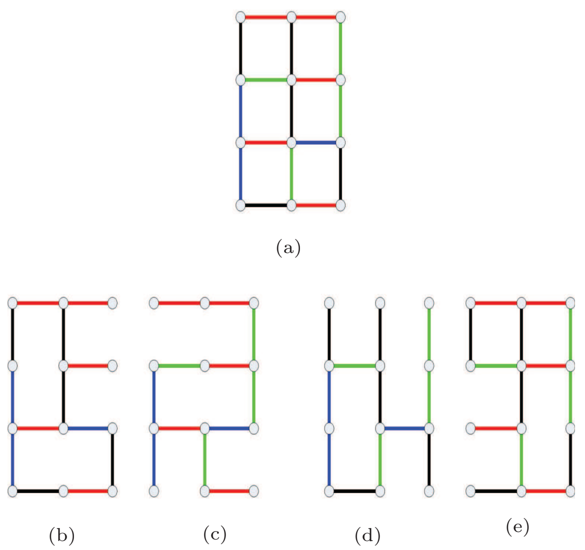


Fig.2. (a) A 2-color connected mesh for $n = 12, m = 17$ and $\gamma = 4$; (b)$\sim$(e) are derived from (a) for $n = 12$ and $\gamma = 3$.

6 Conclusions

Homogeneous faults in the network are prevalent, and most of them are caused by only one reason. Hence, 2-color connected graphs are important and they can be directly used to construct fault-tolerant network topologies. In this paper, we investigated the necessary and sufficient condition for 2-color connected graphs and explored the minimum number of edges in a 2-color connected graph with given parameters, without losing its characteristic. We contributed a general construction method of designing arbitrary 2-color connected graphs. The proposed method is flexible and simple, so that it can be easily applied to any number of vertices and colors. Moreover, when mapping heterogeneous application-specific topologies to a physical NoC topology, the result can be adopted likewise to save time and power.

References

[1] Wang Y, Desmedt Y. Edge-colored graphs with applications to homogeneous faults. *Information Processing Letters*, 2011, 111(13): 634-641.

[2] Wang Y. LT codes for efficient and reliable distributed storage systems revisited. *arXiv: 1207.5542*, 2012. http://arxiv.org/abs/1207.5542, Feb. 2015.

[3] Wang Y. Array BP-XOR codes for reliable cloud storage systems. In *Proc. ISIT*, Jul. 2013, pp.326-330.

[4] Tan S, Lv J. Characterizing the effect of population heterogeneity on evolutionary dynamics on complex networks. *Scientific Reports 4*, 2014, Article No. 5034.

[5] Arora S, Barak B. Computational Complexity: A Modern Approach. Cambridge: Cambridge University Press, 2009.

[6] Constantine G. Colorful isomorphic spanning trees in complete graphs. *Annals of Combinatorics*, 2005, 9(2): 163-167.

[7] Hammadi A, Mhamdi L. A survey on architectures and energy efficiency in Data Center Networks. *Computer Communications*, 2014, 40(1): 1-21.

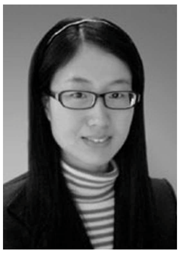
[8] Stensgaard M B, Sparso J. ReNoC: A network-on-chip architecture with reconfigurable topology. In *Proc. the 2nd NoCS*, Apr. 2008, pp.55-64.

[9] Murali S, De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures. In *Proc. DATE*, Feb. 2004, pp.896-901.

[10] Gao Y, Jin Y, Chang Z, Hu W. Ultra-low latency reconfigurable photonic network on chip architecture based on application pattern. In *Proc. OFC*, Mar. 2009.

Rui Hou received his Bachelor's degree in software engineering from Tianjin Polytechnic University (TJPU), Tianjin, in 2012. He is working toward his Master's degree in computer science and technology at the School of Computer Science and Software Engineering in TJPU. His research interests include graph theory, parallel/distributed computing, cluster/grid/cloud computing, optical wireless communication, and high-performance computer architectures.

Ji-Gang Wu received his B.S. degree in computational mathematics from Lanzhou University, Lanzhou, in 1983, and Ph.D. degree in computer software and theory from the University of Science and Technology of China, Hefei, in 2000. He was with the Center for High Performance Embedded Systems, School of Computer Engineering, Nanyang Technological University, Singapore, from 2000 to 2009, as a research fellow. He joined Tianjin Polytechnic University, Tianjin, in 2009, as a Tianjin distinguished professor and the dean of the School of Computer Science and Software Engineering. His research interests include high performance architecture, reconfigurable VLSI design, hardware/software co-design, and combinatorial search.



Yawen Chen obtained her Ph.D. degree in computer science from the University of Adelaide, Australia, in 2008. She received her Bachelor's degree in computer science in 2002 and Master's degree in 2004 from Shandong Normal University, Jinan, and then worked as a researcher in Japan Advanced Institute of Science and Technology in 2005. After her Ph.D. study, she worked as a postdoctoral researcher at Royal Institute of Technology (KTH), Sweden, in 2009. She has been a lecturer in the University of Otago in New Zealand since 2011. Her research interests include resource optimization and performance evaluation in computer networking and computer architecture (e.g., optical networks, interconnection networks, green computing).

Haibo Zhang received his M.S. degree in computer science from Shandong Normal University, Jinan, in 2005, and Ph.D. degree in computer science from the University of Adelaide, Australia, in 2009. From 2009 to 2010, he was a postdoctoral research associate at Automatic Control Laboratory, KTH. Currently he is a lecturer at the Department of Computer Science in the University of Otago, New Zealand. His current research interests include real-time industrial wireless communications, wireless sensor/ad hoc networks, cyber-physical systems, green computing, and distributed algorithms and protocol design.

Xiu-Feng Sui received his B.S. degree in computer science and technology from Harbin Institute of Technology, Harbin, in 2005, and Ph.D. degree in computer architecture from the University of Science and Technology of China, Hefei, in 2010. He has been an associate professor of the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, since 2014. His research interests include high performance computer architectures, system performance modeling and evaluation, and data centers.